НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»

УДК 004.056

«До захисту допущено»
В.о. завідувача кафедри
_____ М.В.Грайворонський
“___”_____2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності: 125 Кібербезпека

на тему: Удосконалені методи автентифікації в системах обміну миттєвими повідомленнями

Виконав (-ла): студент (-ка) 2 курсу, групи ФБ-71мп
(шифр групи)

Лобанов Сергій Олександрович
(прізвище, ім’я, по батькові)

Науковий керівник к.е.н., доц. Ткач Володимир Миколайович _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент к.т.н., доцент ФІОТ Жданова О.Г. _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою
Спеціальність (спеціалізація) – 125 Кібербезпека («Системи і технології кібербезпеки)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)

«___»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Лобанову Сергію Олександровичу

1. Тема дисертації: Удосконалені методи автентифікації в системах обміну миттєвими повідомленнями

науковий керівник дисертації к.е.н., доц. Ткач Володимир Миколайович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «15» листопада 2018 р. № 4171-с

2. Термін подання студентом дисертації 12.12.2018 р.

3. Об'єкт дослідження _____
_____

4. Вихідні дані _____
_____
_____
_____

5. Перелік завдань, які потрібно розробити _____
_____
_____
_____
_____
_____

6. Орієнтовний перелік ілюстративного матеріалу _____
_____
_____

7. Орієнтовний перелік публікацій _____

_____

_____

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка

Студент _____ (підпис) _____ (ініціали, прізвище)

Науковий керівник дисертації _____ (підпис) _____ (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Обсяг роботи 104 сторінки, 2 ілюстрації, 16 джерел літератури, 25 таблиць та 2 додатки.

Об’єктом дослідження є автентифікація користувачів та їх повідомлень на основі виділених поведінкових патернах при спілкуванні в системах обміну миттєвими повідомленнями.

Предметом дослідження є методи багатофакторної автентифікації та способи покращення методів машинного навчання при автентифікації повідомлень користувача за ключовими характеристиками ведення діалогу в системах обміну повідомленнями.

Метою даної кваліфікаційної роботи є підвищення рівня захищеності систем обміну миттєвими повідомленнями за рахунок побудови механізму з більш надійними методами автентифікації користувачів та удосконаленими методами автентифікації повідомлення на основі виділених поведінкових патернів користувача, що дозволить знизити рівень загрози витоку інформації при обміні повідомленнями.

Методами дослідження дипломної роботи є аналіз та порівняння методів автентифікації користувачів та повідомлень на основі поведінкових патернів користувача шляхом аналізу діалогів користувачів у системах обміну миттєвими повідомленнями, побудова механізму перевірки автентичності користувача і повідомлення.

Результатом дипломної роботи є система з двофакторною автентифікацією користувачів та вдосконаленим механізмом автентифікації повідомлень для запобігання витоку інформації в системах обміну миттєвими повідомленнями.

БАГАТОФАКТОРНА АВТЕНТИФІКАЦІЯ, ЗАГРОЗА ВИТОКУ ІНФОРМАЦІЇ, БАЄСІВ КЛАСИФІКАТОР, ВИБІР ПІДМНОЖИНИ ОЗНАК, СИСТЕМИ ОБМІНУ МИТТЄВИМИ ПОВІДОМЛЕННЯМИ

ABSTRACT

The work includes 104 pages, 2 images, 16 links and 25 tables.

The object of research are user authentication and their messages authentication based on extracted behavioral patterns during communication at instant messaging services.

The subject of this qualification is multifactor authentication methods and improvement approaches for machine learning classification methods of user messages authentication by analyzing the characteristics of dialogue in order to use the key features of dialogue in the applications of instant messaging.

The aim of this qualification work is improvement of security of instant messaging systems by building of mechanism with reliable methods of user authentication and improved messages authentication methods for prevention of information leakage based on extracted behavioral patterns of users during communication at instant messaging services.

Methods of research are analysis and comparison of multifactor authentication methods for users and messages based on user behavioral patterns extraction by analyzing of user dialogues during messaging, building of mechanism for user and message authentication.

The result of the work is a system with the multifactor authentication and improved messages authentication information leakage prevention during instant messaging.

MULTIFACTOR AUTHENTICATION, THREAT OF INFORMATION LEAKAGE, BAYESIAN CLASSIFIER, FEATURE SUBSET SELECTION, INSTANT MESSAGING SERVICES

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

IMS – Instant Messaging Service – Система обміну миттєвими повідомленнями

IM – Instant Messaging – Обмін миттєвими повідомленнями

TOTP – Time-based One-time Password – одноразовий пароль на основі часу

HOTP – HMAC-based One-time Password – одноразовий пароль на основі HMAC

HMAC – Hash-based message authentication code – хеш-код автентифікації повідомлень

FSS – Feature Subset Selection - вибір підмножини ознак

QR-код – quick response code - двовимірний код

P (A | B) – умовна ймовірність події A при умові B

Датасет(dataset) — набір даних

ВСТУП

В 2018 році все сильніше розвивається популярність систем обміну миттєвими повідомленнями(Instant Messaging System - IMS), тобто мессенджерам, все більше людей надають перевагу саме їм як основному засобу для обміну інформацією у мережі інтернет. Про їх популярність свідчить постійно зростаюча кількість користувачів у всьому світі, для найбільш розповсюджених IMS станом на жовтень 2018 року вона досягла 1.5 млрд. для WhatsApp, 1.3 млрд для Facebook Messenger, WeChat — 1058 млн., Viber — 260 млн., Telegram — 200 млн. користувачів[1], що в середньому становить майже в півтори рази більше ніж за дослідження січня 2017 року.

Хоча їх популярність і продовжує зростати, такі загрози як використання персональних сторінок або профілів користувачів в цілях отримання конфіденційної інформації або ж примушення до небажаних дій його контактів методами соціальної інженерії залишаються актуальними. Несанкціонований доступ можливий в результаті втрати або викрадення пристрою з встановленими застосунками і збереженими даними для авторизації в них, тимчасового відлучення власника від самого пристрою або ж злому акаунту. В деяких випадках навіть додаткових методів автентифікації користувачів по їх поведінкових патернах при обміні повідомлень може бути недостатньо, тобто стає необхідним застосування удосконалених методів автентифікації користувачів.

Актуальність роботи випливає з постійного зростання розповсюдженості систем обміну миттєвими повідомленнями і задачі запобігання витоку конфіденційної інформації через загрози пов`язані з несанкціонованим доступом до профілів користувачів так і перехопленням повідомлень. В такому випадку звичайних методів автентифікації може бути недостатньо і виникає необхідність застосування нових та вдосконалення існуючих методів автентифікації.

Метою роботи є підвищення рівня захищеності систем обміну миттєвими повідомленнями за рахунок побудови механізму з більш надійними методами автентифікації користувачів та удосконаленими методами автентифікації

повідомлення на основі виділених поведінкових патернів користувача, що дозволить знизити рівень загрози витоку інформації при обміні повідомленнями.

Для досягнення поставленої мети були визначені наступні **завдання**:

- дослідити рівень захищеності сучасних систем обміну миттєвими повідомленнями
- провести аналіз методів автентифікації користувачів, їх перевагами та недоліками при практичному застосуванні
- оптимізувати та удосконалити методи автентифікації особистості відправника повідомлення на основі поведінкових патернів
- реалізувати механізм для автентифікації користувача на початку та під час обміну миттєвими повідомленнями

Об'єктом дослідження є автентифікація користувачів та їх повідомлень на основі виділених поведінкових патернах в системах обміну повідомленнями.

Предметом дослідження є методи багатофакторної автентифікації та способи покращення методів машинного навчання при автентифікації повідомлень користувача за ключовими характеристиками ведення діалогу в системах обміну повідомленнями.

Методами дослідження дипломної роботи є аналіз та порівняння методів автентифікації користувачів та повідомлень на основі поведінкових патернів користувача шляхом аналізу діалогів користувачів у системах обміну миттєвими повідомленнями, побудова механізму перевірки автентичності користувача і повідомлення.

Наукова новизна даної роботи полягає у побудові удосконаленого механізму автентифікації користувача на основі багатофакторної автентифікації та перевірки автентичності повідомлення за його аналізом для запобігання витоку інформації.

Практичне значення результатів роботи полягає у широкому використанні систем обміну миттєвими повідомленнями та високим рівнем загрози витоку інформації, що потребує вдосконалення існуючих методів автентифікації для підвищення рівня захищеності цих систем.

1 ОГЛЯД ЗАГРОЗ ІНФОРМАЦІЇ В СИСТЕМАХ ОБМІНУ МИТТЄВИМИ ПОВІДОМЛЕННЯМИ

Обмін миттєвими повідомленнями (Instant Messaging - IM) став програмним середовищем для спілкування з друзями, членами сім'ї та колегами. Набагато дешевше звертатися до людей через IM, ніж за допомогою традиційних телефонних міжміських дзвінків. В деяких випадках надійність в IM перевершила телефон та інші комунікаційні засоби. IM - це інструмент спілкування в режимі реального часу, який останнім часом став домінуючим над електронною поштою. Особливо в бізнесі, коли важлива транзакція має відбуватися без проблем.

З моменту створення чату, як способу комунікації між товаришами, він перетворився на суттєвий спосіб спілкування для десятків і мільйонів користувачів Інтернету і зростає популярністю як в професійних, так і в персональних додатках. Незважаючи на те, що обмін миттєвими повідомленнями має високу оцінку своєї комунікаційної спроможності, більшість з його реалізацій були розроблені з незначним фокусуванням на безпеку. Популярне використання цих інструментів створило виклики для безпеки як для приватних осіб, так і корпорацій та державних установ.

У додатках обміну миттєвими повідомленнями виявлено численні вразливі місця, що забезпечує легкий механізм застосування різних методів для отримання конфіденційних даних корпорації, прослуховування розмов в чатах, призводить до відмови в обслуговуванні та, в крайніх випадках, до крадіжки особистості користувача. Крім того, IM - це проста платформа для закріплення та розповсюдження черв'яків та вірусів, які потенційно можуть залишатися невиявленими протягом тривалого періоду часу.

На даний час міжмережевим екранам надзвичайно важко блокувати та сканувати порти, що використовуються при IM, або пов'язані з обміном повідомленнями віруси до втручання в внутрішню мережу[2]. Для захисту від загроз, які використовує система державних підприємств IM, адміністратори, а

також кінцеві користувачі повинні знати про ризики безпеки, які вони становлять для себе як індивідів так і своїх товаришів.

Коли ви спілкуєтеся з кимось через сторонню програму для обміну повідомленнями, пам'ятайте, що хтось може прочитати ваші повідомлення, хтось крім вашого призначеного одержувач. На кшталт того як пакет, який залишився без догляду на вашому порозі, ризик того, що хтось отримає доступ до ваших особистих речей або інформації, є високим. Інтернет-конфіденційність є надважливою, тому додатки для мобільних чатів, які заявляють, що бесіди цілком захищені, настільки популярні, незважаючи на деякі недавні насмішки.

Більшість існуючих IMS, включаючи найпопулярніші не можуть вважатися абсолютно безпечними. Вони часто розроблялися та впроваджувались у часи, коли ніхто не думав про безпеку. Тепер, оскільки безпека стає все більш важливою, починають виділятися певні зусилля спрямовані на включення функцій безпеки в ці системи. Тим не менше, більшість досліджених систем виявили серйозні недоліками безпеки.

Багато з них не гарантують, що доступ до інформації буде надаватися лише авторизованим особами. Деякі з них навіть не надають користувачам можливості налаштувати хто зможе бачити їхній профіль. Вони просто розкривають інформацію про їх стан будь-кому хто зацікавлений в цьому. Це притаманно старим версіям деяких месенджерів, у нових версіях користувачі можуть обмежувати доступ до свого статус лише людям зі списку їх контактів. Інші системи дозволяють своїм користувачам визначати свої налаштування щодо наявності сповіщень про їх присутність, але слабка автентифікація та механізми шифрування дозволяють неавторизованим особам мати доступ до їх статусу.

1.1 Рівень захищеності сучасних систем обміну миттєвими повідомленнями

Деякі з систем обміну миттєвими повідомленнями відправляють текст повідомлень у відкритому вигляді, також вони можуть шифрувати лише

повідомлення що йдуть від клієнта до сервера, але використовуючи дуже слабкі алгоритми. Код для шифрування та дешифрування заголовків деяких месенджерів можна знайти в Інтернеті. Таким чином, розшифровка заголовків повідомлень не є проблемою для зловмисника. Хакер може легко перехопити повідомлення, відправлене між клієнтом і сервером, розшифрувати заголовок за допомогою однієї з вищезазначених програм і знайти SESSION_ID користувача, який може бути використаний для підробки повідомлення та відправлення його на сервер. За допомогою зловмисного повідомлення, яке оновлює параметри користувача, злочинець може додати себе до списку людей з дозволом до перегляду інформації про користувача, що призведе до витоку інформації. Даний вид загрози називається спуфінг.

Певні систем обміну миттєвими повідомленнями не можна вважати абсолютно захищеними з двох причин. По-перше, вони не мають належної політики безпеки. Політика безпеки, серед іншого, визначає, які дії можуть виконувати органи системи, і ті, які заборонені. Прикладом поганої політики безпеки є надання кожному доступу до інформації про присутність інших користувачів. По-друге, існуючі месенджери не мають механізмів для забезпечення виконання своїх політик безпеки. Слабка автентифікація забороняє належний контроль доступу. Інформація може поширюватися через незахищені канали, тобто без шифрування, перевірки автентичності та перевірки цілісності. Система обміну миттєвими повідомленнями має задовольняти протоколу RFC2779 який надає вимоги безпеки для моделі обміну миттєвими повідомленнями, визначеною RFC2778 [3].

Розглянемо сучасні IMS та їх рівень захищеності на даний момент.

Facebook Messenger

Facebook Messenger - це система обміну миттєвими приватними повідомленнями від Facebook. Спочатку він був розроблений як Facebook Chat в 2008 році, але був оновлений в 2010 році з автономними випусками, запущений на iOS та Android у 2011 році. Застосування платформи дозволяє користувачам надсилати та отримувати повідомлення, фотографії та відеозаписи з додатковою

підтримкою голосових та відеодзвінків. Також додано наскрізне шифрування на основі інтеграції сторонніх додатків.

Facebook Messenger також може похвалитися деякими цікавими функціями, такими як бот-платформа, запущена в 2016 році та її віртуальний помічник штучного інтелекту, що дозволяє автоматично закінчувати деякі завдання. Facebook Messenger недавно запустив функцію unsend, яка дозволяє користувачам видаляти відправлені повідомлення з поштової скриньки одержувача за перші 10 хвилин за допомогою функції «Видалити для всіх». В даний час ця функція доступна для додатків iOS та Android у деяких країнах, її очікується в найближчому майбутньому.

Тим не менш, користувачі, які освічені в питаннях про конфіденційність та безпеку, можуть захотіти надати цьому і недоліки, - незважаючи на всю популярність Facebook щодо цінової конфіденційності та безпеки, його бізнес-модель по суті побудована на профілях своїх користувачів, збиранні їхніх даних, тому нещодавній скандал Cambridge Analytica, для якого нещодавно було виписано штраф у розмірі 500 000 фунтів стерлінгів від ICO у Великобританії, також не було дуже переконливим. Також, Facebook продає частину цієї інформації стороннім особам, тому додаток залишається безкоштовним. Багато різних послуг використовують дані Facebook як інформацію для авторизації, тому проблема крадіжки особистих даних все ще дуже актуальна.

Skype

Серед багатьох можливостей спілкування та співпраці Skype пропонує функцію обміну миттєвими повідомленнями, інтегровану з додатками Skype та Skype для бізнесу. Це дозволяє користувачам надсилати миттєві повідомлення в режимі реального часу, що також включає в себе відео повідомлення, голосові повідомлення та фото, відео та передачу файлів.

Skype забезпечує додавання шифрування всіх своїх служб та стверджує що використовує TLS (transport-level security - захист на транспортному рівні) для шифрування всіх повідомлень між користувачем Skype та сервісом в хмарі. Він також використовує AES (розширений стандарт шифрування) для всіх

повідомлень, відправлених та отриманих між користувачами Skype. Функція наскрізного шифрування стала доступною для всіх користувачів на платформах iOS, Android, Linux, Mac та Windows у серпні 2018 року.

WhatsApp

Існує три основних принципи співзасновників Whatsapp, які являються його ключовими характеристиками. По-перше, додаток має сприяти приватності та захищати свободу слова. Другий: немає реклами. Третій полягав у тому, що воно повинно бути простим, зручним для користувача. Тим не менше, той факт, що Facebook володіє WhatsApp, не надає впевненості в конфіденційності даних. Facebook, як відомо, має агресивну політику щодо збирання даних про користувачів, також він має певні наміри націлювати рекламу на своїх користувачів базуючись на даних Whatsapp.

Whatsapp розпочав використовувати платформу TextSecure (тепер називається Signal) від Open Whisper Systems у 2015 році, що покращує безпеку, використовуючи справжнє наскрізне шифрування з цілковиту пряму секретність (Perfect Forward Secrecy - PFS). Це означає, що ключі, які використовуються для прокладення зв'язку, не можуть бути захоплені через сервер, а жоден ключ не має доступу до минулих повідомлень.

У квітні 2016 р. протокол Signal був розгорнутий як обов'язкове оновлення для всіх користувачів WhatsApp на всіх мобільних платформах, важливий момент для технології, яка багато років знаходилася на периферіях. Це також зробило Open Whisper Systems найбільш широко використовуваною платформою шифрування на Землі.

У лютому 2017 WhatsApp поступово запровадив двофакторну автентифікацію всім своїм користувачам як необов'язковий додатковий рівень захисту. Двофакторна автентифікація означає перевірку вашої ідентичності двічі - і в цьому випадку користувачі зможуть отримати доступ до свого облікового запису через шестизначний номер. Користувачам WhatsApp потрібно буде увімкнути цю функцію за допомогою їхніх налаштувань, і після ввімкнення,

введення паролю буде залишатися для обраного облікового запису незалежно від того, з якого пристрою він буде використовуватися.

Раніше в 2017 році в звіті Guardian говорилося, що вразливість системи безпеки у WhatsApp означає, що Facebook - головна компанія WhatsApp - може читати зашифровані повідомлення, відправлені через цей сервіс. Дослідник з питань безпеки Тобіас Болтер розповів, що WhatsApp може створювати нові ключі шифрування для офлайн-користувачів, які невідомі відправнику або одержувачу, а це означає, що компанія може генерувати нові ключі, якщо це буде вимагатися.

З точки зору безпеки важливо відрізняти слабо захищені програми для обміну повідомленнями із програм, які мають деякий рівень безпеки. Багато хто використовує шифрування, але працює за допомогою небезпечних каналів, в яких ключі зберігаються централізовано та ховаються за приватними технологіями, які маскують слабкі сторони програмного забезпечення.

Справедливо буде зауважити, що поліція та розвідувальні служби стурбовані покращеним рівнем безпеки який пропонують ці додатки, що може призвести до того, що їм надаватимуть перевагу терористи та злочинці. Щоправда, вони не є неприступними. Використання надійного шифрування забезпечує захист каналу зв'язку, але необов'язково захищає сам пристрій. Існують і інші способи прослуховування повідомлень між людьми, ніж взлом шифрування.

Найновіші додатки, крім обміну повідомленнями, зазвичай являються комбінацією з відео, голосу, чату, обміну файлами, а іноді (хоча з певними складнощами, оскільки мобільні мережі працюють по-різному), обміну повідомленнями через SMS та MMS. Цікава тема полягає в тому, що додатки в цій функції часто використовують технології з відкритим кодом, хоча це не означає, що додатки ідентичні один одному. Інтерфейс користувача та додаткові функції безпеки все одно відрізняються.

Signal

Signal (раніше TextSecure Private Messenger), можливо, є новаторською безпечною платформою для мобільних повідомлень, яка породила абсолютно новий сектор. Ми називаємо це платформою, оскільки Signal - це більше, ніж додаток, який просто розміщується на пристрої Android або iOS та який зберігає ключі шифрування. Сама програма може бути використана для надсилання та отримання захищених повідомлень і вкладених файлів, встановлювати голосові дзвінки та має зручну функцію обміну повідомленнями в групі. Також можна використовувати Signal як додаток за замовчуванням для SMS, але це більше не використовує шифрування за цілим рядом практичних причин і міркувань безпеки. Signal був розроблений як незалежна платформа, яка передає повідомлення через власну інфраструктуру даних.

Використання програми досить просте. Встановлення починається з підтвердження номера телефону, після чого програма буде функціонувати окремо або як програма для обміну повідомленнями за замовчуванням, пропонуючи імпортувати існуючі повідомлення. Найбільш безпечним способом його використання є, мабуть, у ролі додатку для обміну повідомленнями, так що незахищенні повідомлення не надішлються випадково.

Signal базується на протоколі OTR(Off-the-Record Messaging), використовує AES-256, Curve25519 і HMAC-SHA256; захищення голосових повідомлень (раніше програма RedPhone) на основі ZRTP. Цікаво, що в 2017 році Signal додасть зашифровані відеодзвінки до свого списку функцій, посилюючи свій рівень шифрування. Додаток раніше підтримував наскрізне шифрування для голосових дзвінків, але це оновлення забезпечує такий же рівень захищеності відеозв'язку, як і для чату.

Додаткові функції безпеки включають пароль для додатка та блокувальник, який призупиняє відстеження екрану. Також можна контролювати, які типи даних обмінюються Wi-Fi та мобільні мережі. Очевидно, що і відправник, і одержувач повинні мати встановлений додаток, який працюватиме просто за введенним номером телефону будь-якого іншого зареєстрованого користувача.

Telegram

Telegram використовує протокол MTProto, 256-бітове симетричне шифрування AES, шифрування RSA 2048 та обмін захищеними ключами на основі алгоритму Діффі Хелману. Завдяки можливості обробляти широкий спектр вкладень різних типів, він більше схожий на систему хмарних повідомлень, яка замінює електронну пошту, а також захищає обмін повідомленнями для груп до 200 користувачів.

Існує декілька важливих відмінностей між Telegram та іншими програмами, що розглядаються тут, починаючи з того факту, що користувачі можуть знаходитись за логіном користувача, а не тільки за його номером. Це означає, що контакти не завжди повинні знати номер телефону під час використання Telegram, що робить спосіб спілкування ближче до соціальної мережі.

Реєстрація вимагає ввести ім'я користувача на додачу до мобільного номера облікового запису та вимагає, щоб користувач підтвердив номер, отримавши та ввівши SMS-код. Додаток просить доступ до телефонної книги користувача та інших даних, в чому може отримати відмову, і просто повідомити, які контакти в цьому списку вже зареєструвалися в системі.

Платформа також відкрита для використання з негативними намірами, якщо це правильний термін, у тому числі, як повідомляється, використовуються джихадистами в цілях пропаганди, які використовують функціонал для розповсюдження повідомлень. Це не провина розробників, але відображає те, як ці програми можуть бути використані неправильно у ситуаціях, які важко контролювати.

Viber Messenger

Viber це крос-платформний додаток для обміну зашифрованими повідомленнями, який спочатку був доступний на iPhone. Він схожий на Skype. Viber вперше з'явився на платформі Android в 2012 році, а потім на BlackBerry та платформі Windows Phone. У своїй новітній технології шифрування Viber представила сервіс для повного наскрізного шифрування на всіх доступних платформах - Mac, PC, iOS та Android.

Унікальна річ щодо Viber-у, полягає в тому, що вона використовує кодовану систему на основі кольору, щоб показати, наскільки захищена розмова. Сірий колір позначає зашифроване повідомлення. Зелений означає зашифроване спілкування з довіреним контактом, а червоний означає, що існує проблема з ключем автентифікації. Ви також можете приховати будь-які конкретні чати з екрана та отримати доступ до них пізніше.

1.1.2 Загрози витоку інформації та способи їх запобігання

Не дивлячись на сучасний високий технічний рівень захищеності систем обміну миттєвими повідомленнями, актуальними залишаються засоби соціальної інженерії та інші методи які діють не на пряму, обходячи вразливості системи, а через їх користувачів.

Мережеві проблеми: ризик атаки на відмову в обслуговуванні(denial of service) набагато більший за допомогою програми миттєвих повідомлень. Атака на відмову в обслуговуванні заважає законним користувачам отримувати доступ до мережі, використовуючи мережу з високим навантаженням, щоб споживати ресурси, псувати налаштування та змінювати компоненти мережі. Атакуючий здатний перехоплювати повідомлення, налаштовувати пристрої для взаємодії один з одним незаконно, а також використовувати інші ресурси, необхідні для правильної роботи вашої операційної системи.

Шкідливе програмне забезпечення(malware): шкідливі програми, такі як рекламне, шпигунське програмне забезпечення, хробаки, троянські програми та інші віруси, можуть легко передаватися через вашу програму миттєвих повідомлень. Це також включає в себе фішингові програми, які замасковані як легальні, а потім змушують вас розкривати вашу особисту інформацію

Вразливості додатків: точно так само, як вразливості операційної системи ПК, додатки миттєвих повідомлень можуть мати власні вразливі місця для зловживання хакерами. Ситуація ускладнюється через програмне забезпечення, яке підтримує програми миттєвого обміну повідомленнями, що може включати в

себе новий набір вразливостей, які можуть бути успадковані програмою для обміну миттєвими повідомленнями, яка спирається на певний функціонал ОС.

Мобільний IM: збільшення використання бездротових мобільних пристроїв відкриває нові вразливості та проблеми безпеки, які не пов'язані з використанням вашого ПК для систем обімну миттєвими повідомленнями. За допомогою мобільних пристроїв та бездротового з'єднання хакери набагато легше можуть зловживати повідомленнями електронної пошти та учасників вашого списку контактів, визнаючи вразливості, пов'язані з цими додатками.

Фішинг: протягом кількох років фішинг був найпоширенішим напрямом атак проти корпорацій. Хоча завдяки досягненням у їх виявленні та підвищенню рівня обізнаності серед співробітників вони і стають менш ефективними. З цієї причини, і з моменту появи корпоративної політики безпеки, що дозволяє співробітникам приносити власні пристрої на своє робоче місце, традиційні фішингові методи також почали змінюватися та стали більш сфокусованими саме на споживачах за допомогою нових методів та нових векторів розповсюдження.

Сучасні методу фішингу походять від веб-сайтів, соціальних мереж та отримують доступ через мобільні додатки або мобільні веб браузери. Системи обміну миттєвими повідомленнями можуть використовуються для розповсюдження атак через списки контактів своїх жертв. Розповсюдження атаки відбувається швидше, оскільки користувачі читають та реагують на текстові повідомлення в режимі реального часу швидше, на відміну від дій у відповідь на звичайну електронну пошту. Більшість користувацьких програм IMS мають обмежені можливості для фільтрування спаму і ж не мають їх взагалі. Завдяки автоматичному виправленню помилок та невеликих електронних клавіатур на більшості телефонів, промахи в написанні повідомлень та помилки не розглядаються як фактор розгляду автентичності повідомлення.

Фішингові при обміні миттєвими повідомленнями надходять від осіб, які перебувають у списку контактів жертв, і тому вони не розпізнаються. Користувач розпізнає відправника, виявляє довіру до нього, після чого сприймає недостовірну

інформацію як належне. Крім того, отримуючи інформацію від надійного джерела, додаткова перевірка валідності URL-адреси і файла також ігнорується.

У корпоративних середовищах, ефективні цілеспрямовані методи соціальної інженерії включають в себе використання імен впливових керівників та створення нагальної потреби, такі як швидке наближення кінцевого терміну певної задачі. Хакерські компанії орієнтовані на споживачів, часто використовують обмеження часу при відображенні надзвичайно бажаних товарів, що викликає жадібність жертв та сприяє їх помилкам. Одним з таких прикладів є безкоштовні польоти від авторитетних авіакомпаній. Після натискання URL-адреси в повідомленні, жертви потрапляють на фальшиву рекламну сторінку. Ця цільова сторінка містить підроблені елементи спільні до соціальних мереж, такі як велика кількість вподобань під постами та позитивні коментарі від людей, які успішно отримали вигоду від рекламної акції. Метою злочинців у застосуванні цих елементів керування є імітація справжньої соціальної активності. Сповіщення про обмежену і добігаючу кінця кількість вільних місць, може з'явитись на сторінці, щоб посилити почуття нагальності.

Потім жертви залучаються до короткої анкети, щоб виграти акційний приз. Шахраї хочуть залучити користувачів до простих, не вимагаючих набору тексту, ініціюючих питань. Коли користувачі залучаються до процесу, в анкеті запитують до більш суттєвих дій. Типові запити включають пересилання миттєвого повідомлення про рекламу до п'яти друзів, авторизація на веб-сайт певної компанії, створення облікового запису, надання контактної електронної адреси або встановлення промо додатку.

Навчання та перевірка даних необхідні для боротьби з фішингом. Однак, виглядає наче індустрія занадто багато уваги приділяє запобіганню атаки яка постійно змінюється, а не захисту від наслідків, що наступають після атаки. Враховуючи правильний контекст, навіть технічно підкована людина може стати жертвою успішної афери. Замість того, щоб занадто інвестувати в просвітницьку підготовку та отримання обмежених результатів, для запобігання потенційним втратам слід застосовувати такі засоби контролю, як використання

багатофакторної автентифікації (MFA), розділення персональних та робочих пристроїв. [4]

Обмін миттєвими повідомленнями приносить абсолютно новий простір для хакерів, коли йдеться про методи порушення безпеки, оскільки більше людей використовують цю програму для багатьох різних цілей, включаючи передачу файлів. Багато вразливостей було виявлено на ряду з іншими, такими як можливість хакерам легко отримати віддалений доступ до вашого ПК або мобільного пристрою.

Хоча існує безліч переваг використання миттєвих повідомлень, ви повинні мати на увазі, що під час використання цієї програми гарантії захищенності немає. Переконайтеся, що ви користуєтеся всіма перевагами цієї програми та використовуєте її з обережністю.

Наскрізне(end-to-end) шифрування

З огляду сучасних IMS повідомленнями бачимо, що хоча шифрування спілкування було занадто складним підходом для використання в широких колах, але додатки, такі як WhatsApp та Signal ілюструють важливість цього підходу для конфіденційності цифрових даних. Завдяки всім своїм захищеним функціям, таким як зникнення повідомлень та безпечним номерам підтвердження ідентичності, додатки для захищеного обміну повідомленнями можуть по праву надавати вам спокій. Ви напевне повинні використовувати їх. Однак, варто зазначити, що немає такого поняття, як ідеальна безпека.

Наскрізне шифрування перетворює повідомлення на незрозумілі шматки даних, як тільки користувач натискає на кнопку відправки. Звідти повідомлення не відтворюється в щось зрозуміле, поки воно не досягло пристрою одержувача. Попутно це повідомлення нечитаємо, захищене від чужих очей. Це, по суті, як охоронець, який зустрічає вас у вашому домі, їде з вами у вашому автомобілі та проводить до дверей де будь якого місця призначення. Ви в безпеці під час перевезення, але ваша пильність не повинна закінчуватися на цьому.

Незважаючи на те, що наскрізне шифрування є важливою мірою захисту приватності, яке може перешкоджати здійсненню багатьох типів витоку

інформації, ви все одно повинні розуміти інші способи, які атакуючий міг використати для отримання інформації з чату. Навіть коли служба працює ідеально, такі фактори як місце зберігання повідомлень, його отримувачі, а також хто ще має доступ до цих пристроїв, відіграють важливу роль у вашій безпеці. Якщо ви використовуєте додатки шифрованих чатів як один із інструментів для забезпечення конфіденційності та безпеки, це надає вам більше можливостей, в той час покладання на це як на панацею, може нести суттєві ризики.

1.1.3 Вразливості систем обміну миттєвими повідомленнями

Наприкінці квітня 2018 р. дослідження безпеки виявили, що популярний додаток для мобільних повідомлень Viber надсилав відео та зображення без шифрування (еквівалент шифрування повідомлення, таким чином що лише одержувач може розшифрувати і, таким чином, прочитати його). Що ще гірше, додаток також зберігає повідомлення в Інтернеті на загальнодоступному сервері, надаючи можливість отримувати доступ до приватних фотографій та повідомлень для кожного, хто має подібні наміри та знання. Компанія вже вирішила цю проблему, але потенційну шкоду, яку вона завдала, все ще важко визначити.

Snapchat, популярна платформа обміну миттєвими повідомленнями, яка обіцяє видалити кожне повідомлення після перегляду, нещодавно була викрита Федеральною торговою комісією щодо аналогічного порушення. Причина: її повідомлення фактично не зникають так часто, як обіцяли.

У WhatsApp, платформі обміну повідомленнями, яку нещодавно придбала компанія Facebook за 16 мільярдів доларів, у грудні 2018 року була знайдена одна з його вразливостей. Вразливість, про яку йде мова, передбачає використання того самого ключа для декодування шифрування з обох боків розмови, що дозволяє комусь перехоплювати повідомлення, відправлені через Wi-Fi та розшифровувати їх. Зловмисник, який має доступ до зашифрованих повідомлень, може використовувати певний алгоритм для порівняння та суті передбачення тексту, прихованого під шифруванням. У попередніх дослідницьких експериментах криптографи вже використовували цей метод і успішно розшифрували короткі

повідомлення за лічені секунди з точністю до 99%. Оскільки повідомлення, відправлене користувачем на сервер, і навпаки, мають той самий ключ, що їх розшифровує, порівнюючи їх один з одним, фактичний текст може бути витягнутий із зашифрованих потоків. Для тих, хто використовує WhatsApp для надсилання повідомлень з конфіденційною інформацією або просто отримує адресу для зустрічі ввечері, можливість когось побачити текстовий вміст є серйозним ризиком для безпеки для всіх 300 мільйонів щомісячних користувачів.

Поки використання одного і того ж ключа шифрування для захисту двох різних повідомлень являється відомою вразливістю системи безпеки, вона все ще являється трудомісткою задачею для використання. Алгоритм, необхідний для розшифрування перехоплених повідомлень, не тільки складний і забирає багато часу для розробки, але атакуючий повинен мати доступ до бездротової мережі, в якій передаються повідомлення. Незважаючи на це, певний хакер, швидше за все, може створити дещо націлене на користувачів WhatsApp, а також інших вразливих мобільних додатків. WhatsApp обробляє до 27 мільярдів миттєвих повідомлень на день, які, швидше за все, містять велику кількість приватної та потенційно важливої інформації. Однак, незважаючи на здогадки, WhatsApp стверджує, що їх повідомлення є повністю захищеними і незрозуміло, чи продовжують вони вивчати цю проблему на даний момент[5].

Всі ці недоліки в безпеці підкреслюють реальність того, що додатки для мобільних повідомлень не настільки безпечні, як ви вважаєте. Окрім того, чи є дещо насправді анонімним або ефемерним, з того що ви надсилаєте мережею інтернет, чи лишається воно там назавжди в тій чи іншій формі. Насправді після того, як ви натиснете кнопку "опублікувати" або "відправити", а в деяких випадках, навіть раніше, будь-яке повідомлення, фотографія, відео, коментар чи електронна пошта ніколи не будуть справді видалені з інтернету.

Хоча всі ці вразливості є виправними і часто вирішуються швидко після виявлення, тягар захисту приватних повідомлень все ще висить над користувачем. Оскільки більшість із цих програм для мобільних чатів зберігають повідомлення на серверах, перш ніж передавати їх потенційному одержувачу, вони залишають

їх вразливими для розумних злочинців або дір у безпеці. Будь-хто, хто маючи правильні знання, може отримати доступ до цих особистих повідомлень пізніше, без вашої відомості.

1.2 Ознайомлення з методами автентифікації користувачів

На даний момент інформаційні системи (ІС) різного масштабу стали невід'ємною частиною базової інфраструктури держави, бізнесу, громадянського суспільства. Все більше захищається переноситься в ІС. Сучасні інформаційні технології не тільки забезпечують нові можливості організації бізнесу, ведення державної та громадської діяльності, але і створюють значні потреби в забезпеченні безпеки для захисту інформації.

Відомо, що більше 25% зловживань інформацією в ІС відбуваються внутрішніми користувачами, партнерами і постачальниками послуг, що мають прямий доступ до ІС. До 70% з них - випадки несанкціонованого отримання прав і привілеїв, крадіжки і передачі облікової інформації користувачів ІС, що стає можливим через недосконаліть технологій розмежування доступу і автентифікації користувачів. Удосконалення методів системи управління доступом і реєстрації користувачів є одним із пріоритетних напрямків розвитку ІС.

Основними процедурами реєстрації користувачів в ІС є процедура ідентифікації - отримання відповіді на питання «Хто Ви?» і автентифікації - докази того, що «Ви саме той, за кого себе видаєте». Несанкціоноване отримання зловмисником доступу до ІС пов'язано, в першу чергу, з порушенням процедури автентифікації.

Автентифікація - процедура встановлення належності користувачеві інформації в системі пред'явленого ним ідентифікатора. Один із способів автентифікації в інформаційній системі полягає у попередній ідентифікації на основі користувацького ідентифікатора логіна і пароля — певної конфіденційної інформації, знання якої передбачає володіння певним ресурсом в мережі. Отримавши введений користувачем логін і пароль, комп'ютер порівнює їх зі

значенням, яке зберігається в спеціальній захищеній базі даних і, у випадку успішної автентифікації виконує авторизацію з подальшим допуском користувача до роботи в системі.

Традиційну автентифікацію за допомогою пароля називають ще однофакторною або слабкою. Оскільки за наявності певних ресурсів перехоплення або підбір пароля є справою часу. Таким чином часто виникає необхідність використовувати сильну або багатофакторну автентифікацію - на основі двох чи більше факторів. В цьому випадку для автентифікації використостовується не лише інформація відома користувачеві, а й додаткові фактори.

Використання багатофакторної автентифікації для підтвердження особи базується на передумові, що неавторизований користувач навряд чи зможе надати фактори необхідні для доступу. Якщо в спробі автентифікації хоча б один з компонентів відсутній або вказаний невірно, то ідентифікація користувача не встановлюється з достатнім ступенем впевненості та доступу до об'єкту (наприклад, до будівлі або даних), захищеному багатофакторною автентифікацією, залишається заблокованим.

1.2.1 Методи багатофакторної автентифікації

Фактори багатофакторної автентифікації можуть включати:

- деякий фізичний об'єкт яким володіє користувач, такий як USB-накопичувач з секретним токеном, банківська карта, ключ і т. д.
- якийсь секрет, відомий користувачеві, такий як пароль, PIN-код, TAN і т. д.
- деякі фізичні характеристики користувача (біометричні дані), такі як відбиток пальця, райдужка ока, голос, швидкість набору тексту, шаблон в інтервалах натискання клавіш і т. д.
- ваше місцезнаходження, підключення до певної комп'ютерної мережі або використання сигналу GPS для визначення місця розташування.

Фактори знання

Фактори знання є найрозповсюдженішою формою автентифікації. В цьому випадку користувач повинен підтвердити знання певного секрету для автентифікації.

Пароль - секретне слово або рядок символів, які використовуються для автентифікації користувача. Це найбільш часто використовуваний механізм автентифікації. Багато методів багатофакторної автентифікації покладаються на пароль як один з факторів автентифікації. Варіації включають в себе як більш довгі, сформовані з декількох слів (кодові фрази), так і більш коротких, числових, персональних ідентифікаційних номерів (ПІН), що часто використовуються для доступу до банкоматів. Традиційно очікується що паролі будуть зберігатися в пам’яті користувачів

Фактори володіння

Фактори володіння (щось чим володіє користувач і тільки користувач) використовувалися для автентифікації протягом століть, у вигляді ключа до замків. Основний принцип полягає в тому, що ключ уособлює секрет, який поділяється між замком і ключем, цей же принцип лежить в основі автентифікації за фактором володіння в комп'ютерних системах. Токени - це приклад фактора володіння, вони бувають різних типів:

Відключені токени - не мають зв’язку до клієнтського комп'ютера. Зазвичай вони використовують вбудований екран для відображення згенерованих даних автентифікації, які користувач вводить вручну.

Підключенні жетони - це пристрої, які фізично підключені до використовуваного комп'ютера. Ці пристрої автоматично передають дані на комп’ютер. Існує безліч різних типів таких жетонів, включаючи пристрої читання карток, бездротові жетони і USB-токени.

Програмний токен - це тип пристрою двофакторної автентифікації, що може використовуватися для авторизації при використання комп'ютерних сервісів. Програмні токени можуть зберігатися на будь якому електронному пристрої, такому як настільний комп'ютер, ноутбук, КПК або мобільний телефон, і його

можна дублювати. На відміну від апаратних токенів, де облікові дані зберігаються на виділеному пристрої а, отже, не можуть бути дубльовані.

Фактори власності

Характеристикою є фізична особливість суб'єкта. Це може бути портрет, відбиток пальця або долоні, голос або особливість очка. З точки зору суб'єкта, даний спосіб є найбільш простим: не треба запам'ятовувати пароль, ні переносити з собою пристрій автентифікації. Однак біометрична система повинна володіти високою чутливістю, щоб підтверджувати авторизованого користувача, але відкидати зловмисника зі схожими біометричними параметрами. Також вартість такої системи досить велика. Але, незважаючи на свої недоліки, біометрика залишається досить перспективним фактором[6].

Фактор місцезнаходження

Все частіше починає застосовуватися четвертий фактор, який визначається фізичним розташуванням користувача. У той час як користувач підключений до конкретної корпоративної мережі, він може входити в систему використовуючи тільки пін-код, при підключенні до іншої мережі, може вимагатися використання додаткового фактору автентифікації. Це можна розглядати як прийнятний стандарт в якому доступ до офісу знаходиться під контролем.

Системи управління доступом до мережі працюють аналогічним чином, коли ваш рівень доступу до системи може залежати від конкретної мережі, до якої підключено ваш пристрій, наприклад, Wi-Fi чи дротового зв'язку. Це також дозволяє користувачеві переміщатися між офісами та динамічно отримувати однаковий рівень доступу до мережі в кожному з них.

1.2.2 Двофакторна автентифікація

Двофакторна автентифікація(ДФА) є типом багатофакторної автентифікації. ДФА — представляє собою технологію, що забезпечує ідентифікацію користувачів за допомогою комбінації двох різних компонентів.

Багато постачальників різних методів багатофакторної автентифікації пропонують автентифікацію на основі мобільного телефону. Деякі методи

включають в себе автентифікацію на основі повідомлень, QR-коду, одноразових паролей (на основі подій або часу) та перевірку на основі SMS. Перевірка на основі SMS має певні проблеми з безпекою. Телефони можуть бути клоновані, додатки можуть працювати на декількох телефонах, а служби підтримки телефонів можуть мати доступ до тексту SMS. Не менш важливо те, що мобільні телефони можуть бути скомпрометовані в цілому, а це означає, що телефон більше не є тим, що належить лише користувачеві.

Головний недолік автентифікації, включаючи ті методи де користувач володіє чимось, полягає в тому, що користувач повинен практично завжди носити з собою певний фізичний токен (USB-накопичувач, банківську картку, ключ чи щось подібне). Втрата або крадіжка - це ризики. Багато організацій забороняє проносити USB та електронні пристрої в або з приміщення оскільки існує загроза використання шкідливого програмного забезпечення та ризик викрадення даних. Фізичні токени зазвичай не масштабуються, тобто скоріш за все для кожного нового облікового запису та системи буде потрібно виділяти новий токен. Закупівля та подальша заміна токенів такого роду має на увазі в собі додаткові витрати. Крім того, є внутрішні конфлікти та неминучі компроміси між зручністю у використанні та безпекою.

Двофакторна автентифікація мобільних телефонів, що включає такі пристрої, як мобільні телефони та смартфони, була розроблена щоб забезпечити альтернативний метод, який дозволив би уникнути подібних проблем. Щоб автентифікувати себе, люди можуть використовувати свої особисті коди доступу до пристрою (тобто те, що знають лише окремі користувачі), а також одноразові, динамічні пароль, які зазвичай складаються з 4 або 6 цифр. Код доступу може бути відправлений на мобільний пристрій за допомогою SMS, push-сповіщення або може бути згенерований за допомогою додатку для генерації одноразових паролів. У всіх трьох випадках перевага використання мобільного телефону полягає в тому, що немає потреби в використанні додаткового токена спеціального призначення, оскільки користувачі, як правило, постійно тримають свої мобільні пристрої поруч.

1.3 Автентифікація користувачів на основі поведінкових патернів

В попередньому дослідженні була проведена робота над виділенням ключових характеристик, за якими ми маємо можливість встановлювати зв'язок між повідомленням та його автором з певною ймовірністю в IMS. Дані висновки ми можемо зробити завдяки притаманним кожній людині унікальним поведінковим патернам.

Ми вияснили, що поведінкові патерни при обміні повідомленнями можна розглядати як унікальну характеристику самого відправника повідомлення, тобто використовувати у якості автентифікації за вхідним повідомленням. Основним чином, вони виникають через особливість людини діяти згідно звичкам, або ж просто пристосовуватися до навколишнього середовища. Хоча інколи хороші знайомі і можуть відрізнити чи пише хтось інший з профілю його товариша, однак часто цьому може не приділятися відповідної уваги, в таких випадках загроза витоку інформації стає доволі реальною.

Таким чином, може стати у пригоді механізм, що буде брати до уваги певний набір характеристик користувача та аналізувати появу деяких відхилень від попередньо визначеного патерну. Даний підхід дозволяє створити додатковий етап автентифікації користувача та повертатиме певне значення, за яким відправник в більшій або меншій мірі відповідатиме раніше встановленому патерну, визначеному по попереднім діалогам.

Для створення необхідного механізму, що працюватиме при обміні повідомленнями між парою користувачів має використовуватися попередня історія їх листування, оскільки саме за нею можна відтворити в деякій мірі приблизну картину притаманних користувачеві характеристик. Поведінкові патерни користувача при обміні миттєвими повідомленнями можуть мати багато проявів таких, як середній час відповіді на повідомлення, кількість повідомлень за одну відповідь, довжина повідомлень, використання певних слів і фраз при

привітанні або завершенні розмови. Особливістю є і словниковий запас користувача, певні вподобання з вживання додаткових символів або ж смайлів.

Манеру відповідати у певний проміжок часу можемо узагальнити, розділивши це значення на часові проміжки, що відповідатимуть дуже малому часу на відповідь, малому, середньому, великому та іншим. За змістом повідомлення доцільно враховувати словниковий запас користувача та частоту вживання певних слів чи виразів. Таким чином, усі перераховані характеристики можна враховувати при виділенні поведінкових патернів користувачів.

Висновки до розділу 1

В даному розділі було розглянуто рівень захищеності сучасних IMS на основі найпопулярніших месенджерів. Ми ознайомилися з загрозами витоку інформації та способами їх запобігання, виділили випадки коли застосування цих запобіжних методів неможливе або ж певним чином не допомогло перешкодити зловмисному доступу до обміну повідомленнями.

Тобто, виникає необхідність у підвищенні рівня захищеності системи, а саме вдосконалення методів автентифікації для запобігання витоку інформації, можливості використовувати канал зв'язку у зловмисних цілях. По перше, впровадження посиленої автентифікації на початку розмови дозволяє встановити додатковий рівень безпеки. По друге, автентифікація отриманого повідомлення на стороні користувача спеціальним механізмом, який буде аналізувати це повідомлення та сповіщати про можливість порушення його автентичності.

Оскільки подібний механізм було розроблено в попередньому досліджені і нам вдалося досягти певних результатів, розглянемо які методи автентифікації будуть ефективними в досягненні нашої цілі та яким чином ми можемо удосконалити наш існуючий механізм, збільшити його точність та оптимізувати його роботу.

2 МЕТОДИ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ В СИСТЕМАХ ОБМІНУ ПОВІДОМЛЕННЯМИ

Основне призначення систем обміну миттєвими повідомленнями - це проведення розмов через Інтернет. Користувач цієї служби очікує, що його розмови зберігатимуться конфіденційно, тобто ніхто не може їх переглянути. Також він хоче бути впевненим, що повідомлення, які він отримує, це саме ті повідомлення, що були надіслані йому. В кінці кінців, він хоче бути впевненим, що кожне повідомлення було дійсно відправлено стороною, яку він вважає своїм співрозмовником. Тобто кожна з систем обміну миттєвими повідомленнями має відповідати певним правилам:

- тільки назначений приймач (або одержувач) повідомлення повинен мати можливість його прочитати. Не повинно бути перехоплення та прочитання повідомлення;
- користувач, який отримує повідомлення, повинен мати змогу перевіряти, що воно є справжнім, тобто повідомлення не було зміненим, відредагованим або сфабрикований третьою стороною, тобто було автентичним;
- одержувач повідомлення повинен мати можливість перевірити ідентифікацію відправника.

Саме ці питання ми і маємо вирішити, дослідивши способи вдосконалення методів автентифікації користувачів та повідомлень, а саме методи багатофакторної автентифікації та методи аналізу вхідних повідомлень на основі поведінкових патернів користувачів.

2.1 Дослідження методів автентифікації

В основі двофакторної автентифікації лежить використання не тільки традиційної зв'язки «логін-пароль», а й додаткового рівня захисту - так званого

другого фактору, володіння яким потрібно підтвердити для отримання доступу до облікового запису або до інших даних.

Найпростіший приклад двофакторної автентифікації, з яким постійно стикається кожен з нас - це зняття готівки через банкомат. Щоб отримати гроші, потрібна карта, яка є тільки у вас, і PIN-код, який знаєте тільки ви. Отримавши вашу карту, зловмисник не зможе зняти готівку не знаючи PIN-коду і точно так само не може отримати гроші знаючи його, але не маючи карти. За таким же принципом двофакторної автентифікації здійснюється доступ до ваших акаунтів в соцмережах, до пошти та інших сервісів. Багатофакторна автентифікація забезпечує додатковий рівень безпеки, вимагаючи більш ніж одного способу автентифікації, щоб перевірити ідентифікацію користувача для входу або виконання суттєвих транзакцій. Окрім паролів, які історично були одним з факторів, інший фактор може включати те, що ви маєте, наприклад унікальний токен, щось що змінюється кожні 30 секунд (одноразовий пароль на основі часу) або щось таке, як, наприклад, відбиток пальця користувача.

2.1.1 Дослідження та порівняння методів двофакторної автентифікації

Першим фактором є комбінація логіна і пароля, а в ролі другого може виступати один з приведених нижче конкретних методів двофакторної автентифікації, розглянемо їх переваги та недоліки[7].

SMS

Підтвердження за допомогою SMS-кодів працює дуже просто. Ви, як завжди, вводите свій логін і пароль, після чого на ваш номер телефону приходить SMS з кодом, який потрібно ввести для входу в обліковий запис. Це все. При наступному вході відправляється вже інший SMS-код, дійсний лише для поточної сесії.

Переваги:

- Генерація нових кодів при кожному вході. Якщо зловмисники перехоплять ваш логін і пароль, вони нічого не зможуть зробити без коду.

- Прив'язка до телефонного номеру. Без вашого телефону вхід неможливий.

Недоліки:

- При відсутності сигналу мережі ви не зможете залогінитися.
- Існує теоретична ймовірність підміни номера через послугу оператора або працівників салонів зв'язку.
- Якщо ви авторізуетесь і отримуєте коди на одному і тому ж пристрої (наприклад, смартфоні), то захист перестає бути двухфакторной.

U2F Ключі

Універсальний 2-й фактор (Universal 2nd Factor - U2F) - це відкритий стандарт, який використовується з пристроями USB, пристроями NFC та смарт-картками. Щоб автентифікуватися, ви маєте просто підключити його (для USB-ключів), натиснути на нього (для пристроїв NFC) або провести (для смарт-карт) до обладнання.

Переваги:

- Ключ U2F - справжній фізичний фактор. На відміну від SMS-кодів, їх не можна перехопити або перенаправляти. І на відміну від більшості двофакторних методів, ключі U2F є захищеними від фішингу, оскільки вони зареєстровані лише для роботи з конкретними сайтами, на яких ви зареєстровані. Це один з найнадійніших методів 2FA в даний час.

Недоліки:

- Оскільки U2F - відносно нова технологія, вона ще не так широко підтримується. Наприклад, ключі NFC працюють лише з мобільними пристроями Android, тоді як USB-ключі працюють переважно з браузером Chrome (Firefox працює над цим). Ключі U2F також коштують грошей, часто від 10 до 20 доларів, але можуть збільшитися в залежності від того, наскільки міцний ключ ви хочете.

Додатки-автентифікатори

Цей варіант багато в чому схожий на SMS, з тією лише відмінністю, що, замість отримання кодів по SMS, вони генеруються на пристрої за допомогою спеціального додатку (Google Authenticator, Authy). Під час налаштування ви

отримуєте первинний ключ (найчастіше - у вигляді QR-коду), на основі якого за допомогою криптографічних алгоритмів генеруються одноразові паролі з терміном дії від 30 до 60 секунд. Навіть якщо припустити, що зловмисники зможуть перехопити 10, 100 або навіть 1 000 паролів, передбачити з їх допомогою, яким буде наступний пароль, просто неможливо.

Переваги:

- Для автентифікатора не потрібен сигнал мережі, тобто достатньо підключення до інтернету лише при первинному налаштуванні.
- Підтримка декількох акаунтів в одному автентифікаторі.

Недоліки:

- Якщо зловмисники отримають доступ до первинного ключа на вашому пристрої або шляхом злому сервера, вони зможуть генерувати паролі в майбутньому.
- При використанні автентифікатора на тому ж пристрої, з якого здійснюється вхід, втрачається двухфакторність.

Перевірка входу за допомогою мобільних додатків

Даний тип автентифікації можна назвати збірною солянкою з усіх попередніх. В цьому випадку, замість запиту кодів або одноразових паролів, ви повинні підтвердити вхід з вашого мобільного пристрою з встановленим додатком сервісу. На пристрої зберігається приватний ключ, який перевіряється при кожному вході. Це працює в Twitter, Snapchat-і та різних онлайн-іграх. Наприклад, при вході в ваш Twitter-аккаунт в веб-версії ви вводите логін і пароль, потім на смартфон приходить повідомлення із запитом про вхід, після підтвердження якого в браузері відкривається ваша стрічка.

Переваги:

- Не потрібно нічого вводити при вході.
- Незалежність від мережі.
- Підтримка декількох акаунтів в одному додатку.

Недоліки:

- Якщо зловмисники перехоплять приватний ключ, вони зможуть видавати себе за вас.
- Сенс двофакторної автентифікації втрачається при використанні одного і того ж пристрою для входу.

Апаратні токени

Фізичні (або апаратні) токени є самим надійним способом двофакторної автентифікації. Будучи окремими пристроями, апаратні токени, на відміну від всіх перерахованих вище способів, ні при якому розкладі не втратять своєї двохфакторної складової. Найчастіше вони представлені у вигляді USB-брелоків з власним процесором, що генерує криптографічні ключі, які автоматично вводяться при підключенні до комп'ютера. Вибір ключа залежить від конкретного сервісу. Google, наприклад, рекомендує використовувати маркери стандарту FIDO U2F, ціни на які починаються від 6 доларів без урахування доставки.

Переваги

- Ніяких SMS і додатків.
- Немає необхідності в мобільному пристрої.
- Є повністю незалежним девайсом.

Недоліки

- Потрібно купувати окремо.
- Підтримується не у всіх сервісах.
- При використанні декількох акаунтів доведеться носити цілу в'язку токенов.

Резервні ключі

По суті, це не окремий спосіб, а запасний варіант на випадок втрати або крадіжки смартфону, на який мають приходити одноразові паролі або коди підтвердження. При налаштуванні двофакторної автентифікації в кожному сервісі вам дають кілька резервних ключів для використання в екстрених ситуаціях. З їх допомогою можна увійти в ваш акаунт, відв'язати налаштовані пристрої та додати нові. Ці ключі варто зберігати в надійному місці, а не у вигляді скріншоту на смартфоні або текстового файлу на комп'ютері.

Обличчя, голос, відбиток пальців

Розпізнавання обличчя, розпізнавання голосу та сканування відбитків пальців підпадають під категорію біометричних даних. Системи використовують біометричну автентифікацію, коли потрібно, щоб ви дійсно були тими, за кого себе видаєте, часто в областях, де потрібна перевірка безпеки (наприклад, уряд).

Переваги:

- Біометрію надзвичайно важко підробити. Навіть відбиток пальців, який, можливо, найпростіший для копіювання, вимагає певного фізичної взаємодії.
- Розпізнавання голосу потребує певного твердження сказаного вашим голосом
- Розпізнавання обличчя потребує чогось радикального, такого як пластична хірургія. Він не незламний, але досить близький до цього.

Недоліки:

- Найбільший недолік і причина, чому біометрія рідко використовується як двофакторний метод, полягає в тому, що скомпрометований біометричний пристрій може ставити під загрозу саме життя.

2.1.2 Вибір кращого методу двофакторної автентифікації

З попереднього підрозділу бачимо що використання методів двофакторної автентифікації може значно підвищити рівень безпеки вашого акаунту. Багато сервісів за замовчуванням мають SMS верифікацію, надсилаючи коди через текстові повідомлення на ваш телефон, коли ви намагаєтеся ввійти. Але SMS-повідомлення мають багато проблем із безпекою та являються найменш захищеним варіантом для двофакторної автентифікації. Хоча цей спосіб і має недоліки, та важливо пам’ятати, шо використання SMS краще, ніж не використання двофакторної автентифікації взагалі.

Якщо ви не використовуєте двофакторну автентифікацію, для входу в обліковий запис будь кому буде достатньо лише знати ваш пароль. Коли ви використовуєте двофакторну автентифікацію за допомогою SMS, комусь потрібно

буде отримати пароль і отримати доступ до ваших текстових повідомлень, щоб отримати доступ до вашого облікового запису.

Судячи з принципу роботи цього методу автентифікації можна припустити що він є безпечним, однак навіть те, що ваш телефонний номер належить лише вам, цього недостатньо для забезпечення абсолютної захищеності цього способу.

Підміна SIM карт

Якщо хтось знає ваш номер телефону та може отримати доступ до особистої інформації, наприклад останні чотири цифри номера вашого соціального страхування, що на жаль, легко знайти, завдяки численним корпораціям та державним установам, які мали витік даних клієнтів. В такому випадку атакуючі можуть зв'язатися з вашим провайдером та перемістити цей номер телефону на новий телефон. Ця атака відома як "обмін SIM-картами", і це той самий процес, який ви виконуєте, коли ви купуєте новий пристрій і переміщаєте його номер телефону. Людина видає себе за вас, надає особисті дані, і ваша компанія мобільного зв'язку встановлює цей номер на новий пристрій. Тепер зловмисники отримають коди SMS-повідомлення, надіслані на ваш номер телефону на свій телефон. Випадки коли атакуючі підмінюють номер телефону жертви та отримують доступ до їх банківських акаунтів відбуваються доволі часто. В основі цього лежить атака з використанням соціальної інженерії, яка спирається обман вашої компанії мобільного зв’язку.

Перехоплення SMS повідомлення

Також доволі можливо перехопити SMS-повідомлення. Так, наприклад, політичним діячам та журналістам в репресивних країнах варто бути обережними, оскільки уряд може перехопити їх SMS-повідомлення, коли вони надсилаються через телефонну мережу. Це вже траплялося в Ірані, де іранські хакери, як повідомляється, скомпрометували декілька облікових записів Telegram через перехоплення SMS-повідомлень, які надавали доступ до цих облікових записів.

Зловмисники також зловживали проблемами в SS7(Signalling System No. 7), системі з'єднання, що використовується для роумінгу, для перехоплення SMS-повідомлень у мережі та маршрутизації їх в інше місце. Існує багато інших

способів перехоплення повідомлень, у тому числі за допомогою підроблення вежі мобільного зв'язку. SMS-повідомлення не були розроблені для безпечної передачі інформації, тому їх не слід використовувати в цих цілях[8].

Іншими словами, просунуті зловмисники з невеликою кількістю особистої інформації можуть отримати доступ до вашого номеру телефону, щоб отримати доступ до ваших облікових записів, а потім використовувати ці облікові записи, щоб спробувати стягнути банківські рахунки, наприклад. Саме тому Національний інститут стандартів і технологій більше не рекомендує використовувати SMS-повідомлення для двофакторної автентифікації.

Недоліки двофакторної автентифікації у цей спосіб наступні:

- мобільний телефон повинен ловити мережу, коли відбувається автентифікація, інакше повідомлення з паролем просто не дійде.
- ви ділитеся з кимось вашим мобільним телефоном, що впливає на ваше особисте життя і може бути в майбутньому на нього буде приходити спам.
- текстові повідомлення (SMS), які, потрапляючи на ваш мобільний телефон, можуть бути перехоплені.
- текстові повідомлення приходять з деякою затримкою, так як деякий час йде на перевірку.
- сучасні смартфони використовуються як для одержання пошти, так і для отримання SMS. Як правило електронна пошта на мобільному телефоні завжди включена. Таким чином, усі акаунти, для яких пошта є ключем, можуть бути зламані (перший фактор). Мобільний пристрій (другий фактор). Висновок: смартфон змішує два чинника в один.

Таким чином, найкращим способом стає генерація паролей на вашому пристрої. Хорошим прикладом двофакторної автентифікації є авторизація Google і Microsoft. Коли користувач заходить з нового пристрою, крім автентифікації по імені та паролю, його просять ввести шестизначний (Google) або восьмизначний (Microsoft) код підтвердження. Ви можете отримати його за допомогою SMS, або голосового дзвінка на ваш телефон, він може бути взятий з заздалегідь складеного реєстру разових кодів або ви можете використовувати додаток-автентификатор,

генеруючий новий одноразовий пароль за короткі проміжки часу. Вибрати один з методів можна в налаштуваннях вашого Google або Microsoft-акаунта. Перевагами двофакторної автентифікації за допомогою цього методу наступні:

- не потрібні додаткові токени, тому що мобільний пристрій завжди під рукою.
- код підтвердження постійно змінюється, а це безпечніше, ніж однофакторний логін-пароль

Зараз майже всі великі сервіси, такі як Microsoft, Google, Yandex, Dropbox, Facebook, вже надають можливість використовувати двофакторну автентифікацію. Причому для всіх з них можна використовувати єдиний додаток автентифікатор, що відповідає певним стандартам, такі як Google Authenticator, Microsoft Authentificator, Authy або FreeOTP.

2.1.3 Дослідження автентифікації за допомогою одноразового паролю згенерованого на основі часу

Для досягнення нашої цілі розглянемо алгоритм створення одноразових паролів за часом(Time-based One-Time Password algorithm - TOTP, визначений у RFC 6238[9]). Він заснований на алгоритмі одноразового пароля на основі HMAC (HOTP, RFC 4226[10]), який використовує не точне зазначення часу, а поточний інтервал з встановленими заздалегідь межами (наприклад, 30 секунд).

Алгоритм діє так: клієнт бере поточне значення таймера і секретний ключ, хешує їх за допомогою якої-небудь хеш-функції і відправляє серверу, у свою чергу сервер проводить ті ж обчислення після чого йому залишається тільки порівняти ці значення. Він може бути реалізований не тільки на хеш-функції SHA-1, на відміну від HOTP, тому хеш-функція також є вхідним параметром.

Аналіз захищеності (RFC 4226) цього алгоритму робить висновок, що найкращою атакою проти функції HOTP є атака грубої сили(brute force). Захищеність алгоритму може бути приблизно оцінюватися так:

$$\square = \frac{\square\square}{10^{\square}} \quad (2.1)$$

де s - вікно синхронізації паролей, v - кількість спроб перевірки пароля, d - кількість цифр згенерованого TOTP пароля.

RFC 4226 рекомендує, мати довжину згенерованого паролю не менше 6 символів, що і вживається такими генераторами одноразових паролей як Google Authenticator, Authy та іншими. Можна використовувати паролі розміром до 9 цифр, що є максимальним значенням, яке може бути представлено 31-бітним рядком.

Оцінка загрози в такому випадку становить, при s = 10 і d = 6 становить

$$\square = \frac{\square}{10^{5}} \quad (2.2)$$

тобто v має становити 10.000 спроб, для досягнення успіху хакером.

Однак, обмежуючи кількість невдалих спроб автентифікації, які користувач може використати за певний проміжок часу, можна досягти чудового значення p, що задовольнятиме вимогам безпеки.

Існує багато додатків для мобільних пристроїв які реалізує алгоритм TOTP. Вони дозволяють створювати тимчасові паролі, які можуть бути використані для авторизації користувачів на сервері автентифікації, який використовує таємний ключ користувачів. Наприклад Google Authenticator використовується переважно для доступу до служб Google, використовуючи двофакторну автентифікацію, однак його можна використати і в розробці рішень автентифікації для web додатків.

Алгоритм TOTP використовується для визначення того, чи справді користувач вказує достовірний пароль, визначений на основі на спільного секретного часу та часової мітки.

Створення облікових даних - це процес, в якому сервер спильний секретний ключ і ділиться ним з клієнтом. Спільний таємний ключ створюється за допомогою сильного криптографічного генератора псевдовипадкових чисел. Далі ключ повідомляється клієнту, щоб він міг налаштовувати свій токен. Найпоширеніший спосіб передачі цієї інформації - це відправка його клієнту у вигляді закодованого QR-коду. Користувач має лише сканувати QR-код за допомогою програми Google Authenticator або іншої, а потім видалити цей код. З метою забезпечення безпеки генератори токенів TOTP зазвичай не дозволяють користувачам отримувати спільні таємні ключі після налаштування облікового запису. Таким чином, лише власники мобільних пристроїв, на яких відбувалися налаштування, які володіють токеном, можуть використовувати цей ключ для створення паролів TOTP.

Автентифікація облікових даних - це процес, в якому сервер застосовує алгоритм генерації TOTP до спільного ключа та паролю, введеного користувачем, для підтвердження його автентичності. Створення спільного ключа є важливим процесом, тому що здатність перехопити чи вгадати чужий ключ може призвести до викрадення облікового запису. Таким чином, варто використовувати сильні генератори псевдовипадкових чисел.

Програма Google Authenticator може бути швидко налаштована за допомогою QR-коду: програма запитує користувача сфотографувати код, а програма використовує дані, закодовані в ньому, для налаштування нового облікового запису. Цей підхід має кілька переваг: людські помилки зведені до мінімуму або взагалі усуваються, процес встановлення простий та швидкий, але найголовніше, загальний секретний ключ ніколи не відображається в звичайному тексті. Якщо злочинцеві не вдасться викрасти використане зображення QR-коду, щоб налаштувати інший обліковий запис автентифікатора, то спільний таємний ключ не буде доступним для прочитання навіть для самого законного власника. Тому QR-код значно зменшує ризик перехоплення під час фази початкової взаємодії.

Щоб перевірити пароль, алгоритм TOTP вимагає:

• Пароль для перевірки.

• Загальний секрет.

• Часова фактор руху.

Клієнт та сервер повинні узгодити спосіб обчислення фактору зсуву на основі часу. RFC 6238[9] рекомендує використовувати розмір за замовчуванням 30 секунд.

Таким чином, принцип роботи автентифікації на основі одноразового згенерованого паролю буде наступний:

1. Генерується секретний ключ на основі якого створюється QR код що показується користувачу
2. Користувач сканує даний QR код власним додатком Google Authenticator
3. Йому відображається згенерований TOTP який необхідно ввести протягом певного часу в додаток для підтвердження особистості
4. Введений пароль перевіряється за допомогою секретного ключа

По суті, TOTP є варіантом HOTP алгоритму, в якому в якості значення лічильника підставляється величина, що залежить від часу. Позначимо:

T— дискретне значення часу, що використовується в якості параметра.

X— інтервал часу, протягом якого дійсний пароль.

T_0— початковий час, необхідний для синхронізації сторін.

K— спільний секрет.

$CurrentTime$— поточний час.

Тоді

$$T = (CurrentTime - T_0)/X \tag{2.3}$$

$$\text{HOTP}(K,T) = Truncate(\text{HMAC-SHA-1}(K,T)) \tag{2.4}$$

$$\text{TOTP} = \text{HOTP}(K,T) \tag{2.5}$$

де

- HMAC-SHA-1(K,T) - генерація 20-ти байт на основі таємного ключа і часу за допомогою хеш-функції SHA-1.
- Truncate — функція вибору певним способом 4 байт:

позначимо String — результат HMAC-SHA-1(K,T); OffsetBits — молодші 4 біта рядка String; Offset = StringToNumber(OffsetBits) і результатом Truncate буде рядок з чотирьох символів — String[Offset]...String[Offset + 3]

Також варто відзначити, що на відміну від HOTP, який заснований тільки на SHA-1, TOTP може також використовувати HMAC-SHA-256, HMAC-SHA-512 та інші HMAC-хеш-функції[3]:

$$\mathrm{TOTP}(K,T) = Truncate(\mathrm{HMAC\text{-}SHA\text{-}256}(K,T)) \tag{2.6}$$

Отже, автентифікація користувача на основі TOTP може слугувати надійним способом автентифікації, оскільки дозволяє переконатися у тому, що саме співрозмовник має безпосередній доступ до необхідного мобільного пристрою. Також, на початку спілкування за відсутності великої кількості даних, достатніх для розпізнавання поведінкового патерну користувача за новими повідомленнями, даний метод являється основним методом запобігання витоку конфіденційної інформації.

За словами прихильників, багатофакторна автентифікація може значно знизити випадки онлайн крадіжок ідентичності інших онлайн злочинів, оскільки лише паролю жертви вже не буде достатнім, щоб дати злодієві постійний доступ до важливої інформації. Тим не менш, методи багатофакторної автентифікації все ще залишаються вразливими для фішингу, атак людини в браузері (англ. man-in-the-browser) і людина посередині(англ. man-in-the-middle).

Таким чином, навіть при наявності такого надійного механізму, як багатофакторна автентифікація, загрози витоку інформації все ж можуть залишатися актуальними, тому необхідно забезпечити додаткові методи для їх

запобігання, а саме, застосування додаткового рівня автентифікації користувача за надісланими повідомленнями.

2.2 Удосконалення аналізу поведінкових патернів за допомогою методів машинного навчання

В ході попередніх досліджень для вирішення нашої задачі були виділені ключові характеристики для тренування різних методів машинного навчання, та проведений їх детальний аналіз та порівняння. Оскільки для створення необхідного механізму, що працюватиме при обміні повідомленнями між парою користувачів використовується попередня історія їх листування, то саме на ній і будувався набір даних для тренування наших методів.

2.2.1 Результати попередніх досліджень

Поведінкові патерни користувача при обміні миттєвими повідомленнями можуть мати багато проявів таких, як середній час відповіді на повідомлення, кількість повідомлень за одну відповідь, довжина повідомлень, використання певних слів або фраз при привітанні або завершенні розмови. Особливістю є і словниковий запас користувача, певні вподобання з вживання додаткових символів або ж смайлів.

Манеру відповідати у певний проміжок часу можемо узагальнити, розділивши це значення на часові проміжки, що відповідатимуть дуже малому часу на відповідь, малому, середньому, великому та іншим. За змістом повідомлення доцільно враховувати словниковий запас користувача та частоту вживання певних слів чи виразів. Таким чином усі перераховані характеристики, що відповідають певним поведінковим паттернам, тому вони і враховувалися при вирішенні нашої задачі.

При аналізі повідомлень було досліджено та порівняно наступні методи класифікації машинного навчання: k-Найближчих сусідів k-NN, штучна нейронна мережа aNN, Метод Опорних Векторів, баєсів класифікатор. За результатами

дослідження виявили, що найефективнішим методом для цієї задачі являється Баєсів класифікатор.

Баєсів класифікатор

Припустимо, ми знаємо певну особливість обміну повідомленнями, яка є типовою для деякого користувача, наприклад, одного друга, на основі якої по вхідному повідомленню від нього ми можемо судити про те, чи належить воно йому з певною ймовірністю. Ця проста ідея може бути узагальнена шляхом застосування методів в теорії ймовірності.

Наприклад, для певного користувача можемо розділити набір повідомлень на два класи: ті що належать йому, та ті що не належать, і існує деякий ймовірнісний розподіл повідомлень (точніше векторів ознак, за допомогою якого описуємо повідомлення) відповідно для кожного класу: P (x | c) визначає ймовірність появи повідомлення з вектором ознак **x** з класу **c**. Зазвичай нам вже відомо дещо про ймовірнісний розподіл для цих класів(як у наведеному вище прикладі, ми знаємо, що ймовірність отримання повідомлення з певною рисою одного користувача, наприклад використання особливих додаткових елементів, буде зустрічатися у інших користувачів з меншою частотою).

Таким чином те, що ми хочемо дізнатися — це чи належить повідомлення x, за деякими його особливостями, користувачеві з профілю якого воно було надіслане за відомим попередньо визначеним поведінковим патерном, що відповідатиме частоті появи конкретних ознак притаманних цьому користувачу при обміні повідомленнями. Тобто, формулюючи нашу задачу у термінах теорії ймовірності, ми хочемо знайти ймовірність $P(c \mid x)$, що за теоремою Баєса обчислюється за формулою (2.7):

$$P(c \mid x) = \frac{P(x \mid c)}{P(x)} = \frac{P(x \mid c)P(c)}{P(x \mid c)P(c)+P(x \mid \bar{c})P(\bar{c})} \qquad (2.7)$$

де $P(x)$ позначає апріорну ймовірність повідомлення x і $P(c)$ - апріорної ймовірності класу c (тобто ймовірність того, що випадкове повідомлення

належатиме до цього класу). Тому, якщо нам відомо значення ймовірностей *P(c)* і *P(x | c)* (для $C \in \{S, D\}$), можемо визначити, *P (c | x)*, на чому і базуватиметься основний правило роботи нашого класифікатора [11]:

Якщо *P(D | x) > P(S | x)* (тобто, апостеріорна ймовірність того, що x не належить нашому відправнику більша, ніж апостеріорна ймовірність того, що x йому належить), тоді відносимо отримане повідомлення до небезпечних, про що повідомляємо користувача, що використовує подібний застосунок.

Також це правило називається правилом максимуму апостеріорної ймовірності. Застосовуючи теорему Баєса можемо звести наше правило до наступного вигляду:

Якщо $\frac{P(x \mid D)}{P(x \mid S)} > \frac{P(D)}{P(S)}$, тоді класифікуємо повідомлення x як небезпечне, інакше з більшою ймовірністю відносимо його до безпечного [12].

Таким чином, основне правило класифікації можна подати у вигляді:

$$P(x) \lessgtr \frac{P_D}{P_S} \tag{2.8}$$

Де повідомлення відноситиметься до небезпечного при справджуванні знаку >, та до безпечного <.

Однак під час подібних обрахунків потрібно враховувати і можливість хибного визначення класу повідомлення класифікатором. Для цього введемо $L(c_1, c_2)$, що позначатиме ризик неправильної класифікації елементу класу c_1 як елементу, що належить до класу c_2 (хоч і вірним було б припускати, що *L(S, S) = L(D, D) = 0*, однак в більш загальному випадку це припущення може бути не завжди вірним). Тоді, очікуваний ризик класифікації даного повідомлення x до класу c буде визначатися за формулою (2.9):

$$R(c \mid x) = L(c, S)P(c \mid x) + L(c, D)P(c \mid x) \tag{2.9}$$

Цілком очевидно, що ми хочемо щоб наш класифікатор мав невеликий ризик при визначенні кожного повідомлення, тому доцільно використовувати наступне правило для класифікації:

Якщо $R(S \mid x) < R(D \mid x)$, тоді класифікуємо повідомлення x як небезпечне, інакше відносимо його до незагрожуючого. Це правило називається правилом класифікації Баєса (або баєсовим класифікатором).

Тепер розглянемо наївний баєсів класифікатор, як ймовірнісний класифікатор, що використовує теорему Баєса для визначення ймовірності приналежності спостереження (елемента вибірки) до одного з класів при припущенні (наївному) незалежності змінних.

Наївний баєсів класифікатор

Тепер, коли ми дослідили теоретичні відомості про оптимальний класифікатор з використанням теореми Баєса, перейдемо до можливості практичної реалізації даної ідеї. Щоб побудувати баєсів класифікатор для виявлення підозрілих повідомлень згідно виділеного поведінкового патерну користувача, ми повинні визначити ймовірності $P(x \mid c)$ і $P(c)$ для будь-яких x і c.

Наприклад, ймовірність $P(S)$ може бути апроксимована за відношенням кількості повідомлень що належать певному користувачу, до кількості всіх повідомлень з тренувального набору даних. Оцінка $P(x \mid c)$ буде набагато складнішою, оскільки її значення фактично залежить від того, як саме ми виберемо вектор ознак для повідомлення m. Давайте розглянемо найпростіший випадок вектору ознак, який буде мати лише єдиний бінарний атрибут, що описуватиме наявність певного слова w в повідомленні. Тобто, ми визначаємо вектор ознак для повідомлення x_w, в якому, для загального набору слів що ми визначимо, 1 означатиме наявність слова w в повідомленні, а 0 - відсутність. У цьому випадку оцінити необхідну ймовірність з наших даних можна наступним чином (2.10):

$$□(□_□ = 1 \mid □) \approx \frac{□_□}{□} \qquad (2.10)$$

Де $□_□$ - кількість повідомлень користувача, в яких міститься слово *w*, а *N* — загальна кількість повідомлень цього користувача [13].

Таким чином, для прийняття рішення по класифікації повідомлення за допомогою теореми Баєса, все що нам потрібно — враховувати наявність слова w, або ж іншого атрибуту, у повідомленні конкретного користувача. Можемо підвести підсумки особливостей використання для даного класифікатору:

Тренування:

1. Обчислюємо $P(S)$, $P(D)$, $P(x_w = 1 \mid c)$, $P(x_w = 0 \mid c)$ (для $c = S, D$) з набору даних для тренування.

Класифікація:

1. Для отриманого повідомлення *m* визначимо вектор ознак x_w і обчислимо ймовірності належності до обраних класів та відносимо його до більш ймовірного класу [11].

На даний момент цей класифікатор навряд чи буде ефективним, оскільки його рішення залежить лише від наявності або відсутності деякого слова в повідомленні. Ми можемо значно покращити цю ситуацію, за допомогою розширення нашого вектору ознак, додаючи до нього більшої кількості атрибутів. Задамо кілька слів $w_1, w_2, \ldots, w_m$ і визначимо для повідомлення m його вектор ознак $x = (x_1, x_2, ..., x_m)$, де x_i дорівнює 1, якщо слово w_i присутнє в повідомленні, і 0 в іншому випадку. За алгоритмом, зазначеним вище, ми повинні обчислювати і зберігати значення $A(x)$ для всіх можливих значень *x* (тобто їх було б 2^m).

Однак, на практиці подібна реалізація була б неможливою для всіх існуючих слів, тому висуваємо додаткове припущення: нехай компоненти вектору *x* являються незалежними в кожному класі. Тобто, присутність одного з слів w_i в повідомленні не впливає на ймовірність присутності інших слів в цьому повідомленні. Хоча дане припущення і не вірне, однак воно дозволяє обраховувати потрібні ймовірності без необхідності зберігати великих обсягів даних, оскільки за незалежності слів маємо(2.11):

$$□(□ \mid □) = \prod_{□=1}^{□} □_□(□_□) \qquad (2.11)$$

Таким чином, запропонований вище алгоритм, легко перетворюється на наївний баєсовий класифікатор. Слово «наївний» у назві означає наївність припущення про незалежність змінних кожного класу. Цікаво зазначити, що даний алгоритм набув широкого використання серед фільтрації спаму та добре зарекомендував себе на практиці

Тренування:

1. Для всіх w_i знайти та зберегти $\Lambda_i(x_i)$ для $x_i = 0, 1$. Обчислити $P(S)/P(D)$.

Класифікація:

1. Визначити вектор ознак x, обчислити $\Lambda(x)$ за добутком збережених змінних $\Lambda_i(x_i)$. Класифікувати вхідне повідомлення [13].

Перевагою цього підходу є те, що вимоги до розміру вибірки скорочуються від експоненційних до лінійних. Також, для отримання результату, класифікатор має лише обчислити значення ймовірностей для нового елементу, завдяки чому його продуктивність не зменшується з збільшенням розмірів датасету. Недоліком моделі можна вважати те, що вона точна лише у випадку, коли виконується припущення про незалежність.

В іншому випадку, строго кажучи, обчисленні ймовірності вже не є точними (і навіть більше того, їх сума може не дорівнювати одиниці, через що потрібно нормувати результат). Однак на практиці незначні відхилення від незалежності призводять лише до незначного зниження точності, і навіть у разі істотної залежності між змінними результат роботи класифікатора продовжує корелювати з істинною приналежністю елементу до класам. При цьому переваги класифікатора (висока швидкість роботи, простота і масштабованість, помірні вимоги до пам'яті) часто переважують недоліки.

Основними характеристиками, які нам важливі при виборі класифікатора були його швидкість роботи, можливість почати працювати за малим набором даних, здатністю перенавчатися. Також, щоб його робота базувася на принципі побудови моделі рішень, тобто не потребує зберігати всі елементи з набору даних для тренування у пам’яті, що суттєво в плані ресурсів для практичного

застосування. Ми виділили основні властивості з отриманих повідомлень користувача що формуватимуть його поведінковий патерн та визначити чіткі критерії у його розпізнавання для точної роботи нашого класифікатора.

2.2.2 Просунутий аналіз поведінкових патернів

Враховуючи що наш алгоритм тренується та виконує класифікацію базуючись на обраному векторі ознак, доцільно буде розглянути варіанти його зміни і розширення. При попередньому дослідженні враховувалися лише слова, символів та розподіл повідомлень за часом, що надає нам можливість отримати непогані результати. Однак, більш поглиблений аналіз обраних характеристик показує що при тренуванні і тестуванні нашого механізму враховувалися не всі особливості написання повідомлень, такі як правопис слів, схильність до порушень певних правил, що може в значній мірі вплинути на точність класифікації.

Для цієї задачі потрібно використати алгоритм розпізнавання помилок та їх виправлення. В цьому нам допоможе орфографічний коректор Норвіга[14], який класифікує типи помилок та виправляє їх з доволі високою точністю незважаючи на свою простоту. Даний коректор допомагає вибрати правильне виправлення з декількох припущень використовуючи значення ймовірностей з якою певне припущення вірне:

$$p = \operatorname{argmax}_c P(c|w) \tag{2.12}$$

Згідно з теоремою Баєса - даний вираз може бути записаний у формі еквівалентній до наступного виразу:

$$p = \operatorname{argmax}_c P(w|c)\, P(c) / P(w) \tag{2.13}$$

Оскільки P(w) рівноймовірна для всіх c, то ми можемо відкинути P(w), що дасть нам:

$$p = \text{argmax}_c \, P(w|c) \, P(c) \tag{2.14}$$

Цей вираз складається з трьох частин:

1. P(c) – ймовірність появи певного слова c (частота вживання c). Ця ймовірність обумовлена самою мовою, тоніше її моделлю. Інакше кажучи, P(c) визначає як часто в тексті певної мови зустрічається певне слово. Для деяких слів P буде доволі високою, при тому як P(“пацифікаційний”) буде меншою, а P(“виліаутаауіа”) буде близько нуля.
2. P($w|c$) – ймовірність того, що автор помилився та написав w, хоча мав на увазі c. Ця ймовірність обумовлена частотою тих чи інших помилок в мову, і називається моделлю помилок мови.
3. argmax_c – оператор, який перебирає всі можливі c в пошуку найімовірніших з можливих кандидатів, тобто шукає таке допустиме c, яке максимізує умовну ймовірність появи w при даному c.

Розглянемо відповідь на можливе питання щодо перетворення простого виразу “argmax_c P($c|w$)” в більш складний вираз, в якому використовуються аж дві мовні моделі, замість однієї. Справа в тому що P($c|w$) враховує в собі відразу обидві мовних моделі, тому очевидно, що простіше виділити ці моделі і працювати з ними окремо. Припустимо що у нас є слово з помилкою – “капка”, це може бути як “шапка”, так і “крапка”. Обидва виправлення мають приблизно однакову частотність в українськії мові. Також, можна враховувати розміщення літер в українській розкладці клавіатури, однак це також не може слугувати повним доведенням правильності одного слова над іншим. Тому краще не розглядати P($c|w$) як єдину величину, бо нам доводиться враховувати і частоту виправлення c і ймовірність виправлення c для даної помилки в w. Зручніше працювати з цими двома ймовірностями окремо.

Таким чином, для тренування подібної моделі знадобиться великий набір даних в якому можна зустріти фактично природній розподіл слів мови, обраної для тестування. На ньому можна буде навчати дану стохастичну модель, а саме підраховувати частоту появи кожного слова в обраному тексті.

Однак при такому підході ми стикаємося з проблемою - що робити зі словами, яких не було при тренуванні. Було б неправильним заявляти, що такі слова мають ймовірність 0, тільки тому, що вони не були присутні в нашій вибірці. Взагалі кажучи, є кілька варіантів розв'язання проблеми. Зараз ми виберемо найпростіше - будемо вважати, що ми «бачили» будь-яке слово хоча б один раз, навіть якщо цього слова не було в нашій файлі. У статистиці такий підхід називається згладжуванням - ми як би піднімаємо провали на графіку розподілу ймовірностей нашої моделі мови.

Тепер звернемо увагу на проблему перерахування всіх можливих виправлень c для слова w. Часто говорять про відстань (edit distance) між двома словами, маючи при цьому на увазі число символів, які треба змінити в одному з слів, щоб слова стали ідентичними. Зміною може бути видалення символу, транспозиція символів (коли символи міняються місцями), заміна одного символу іншим або вставка нового символу. Вважається, що від 80% до 90% всіх помилок знаходяться від оригіналу в одному кроці (в сенсі описаної вище відстані). Для перевірки слів які знаходяться в двох кроках від початкового слова необхідно провести повторну перевірку для кожного з можливих варіантів після першого етапу, по статистиці 98,9% всіх помилок належать саме цій дистанції, тобто подальші перевірки робити майже немає сенсу.

Для нашої задачі нам необхідно враховувати саме схильність користувача помилятися тим чи іншим чином, тому додамо до нашого вектору ознак нашого повідомлення наступні параметри відповідно типу помилки: видалення літери. вставка, заміна і перестановка місцями, також врахуємо чи знаходиться виправлення на відстані двох виправлень від слова з помилкою, або ж виправлення не знайшлося взагалі.

2.2.3 Удосконалення вибору характеристичного вектору

Важливим фактором що впливає на продуктивність роботи будь якого методу машинного навчання являється набір характеристик які ви використовуєте при навчанні. Таким чином, невідповідні, або слабо відповідні характеристики можуть негативно вплинути на продуктивність вашої моделі.

Вибір ключових характеристик(Feature Subset Selection - FSS) - це процес, в якому ви автоматично вибираєте ті характеристики у ваших даних, які найбільшою мірою сприяють прогнозуванню або результату роботи вашого алгоритму. Маючи недоречні характеристики, серед ваших даних, ви можете зменшити точність багатьох моделей, особливо лінійних алгоритмів, таких як лінійна та логістична регресія.

Методики вибору ознак слід відрізняти від виділення ознак. Виділення ознак створює нові ознаки з функцій від первинних ознак, тоді як вибору ознак повертає підмножину ознак. Методики вибору ознак часто використовуються в тих областях, де є багато ознак, і порівняно мало зразків

Завдяки додатковому вибору характеристик перед моделюванням даних можливо досягнути наступних покращень:

- зменшити перенавчання: менша кількість надлишкових даних надає менше можливостей приймати рішення на основі шумів.
- покращити точність: зменшення кількості неінформативних даних дозволяє збільшити точність побудованих моделей.
- зменшити час тренувань: менша кількість даних означає, що алгоритми тренуються швидше.

Розглянемо наступні методи вибору підмножини характеристик у машинному навчанні:

Одномірний вибір

Статистичні тести можуть бути використані для вибору найвпливовіших характеристик на відношення між вхідними та вихідними значеннями.

Одномірний вибір характеристик (Univariate Selection) аналізує кожну характеристику окремо і визначає наскільки міцний взаємозв'язок між нею та вихідним результатом. Ці методи являються простими для виконання і розуміння, в цілому вони особливо корисні для кращого розуміння даних. Існує багато різних методів одномірного вибору.

Один з найпростіших методів розуміння співвідношення характеристики до значення вихідного результату - це коефіцієнт кореляції Пірсона, який обчислюється за формулою(2.15):

$$r_{xy} = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{x})^2 \sum_{i=1}^{m}(y_i - \bar{y})^2}} = \frac{cov(x,y)}{\sqrt{s_x^2 s_y^2}} \quad (2.15)$$

де $x^m = (x_1, \cdots, x_m)$, $y^m = (y_1, \cdots, y_m)$- вибірки

$\bar{x}, \bar{y}$ - вибіркові середні

x^m і y^m, s_y^2, s_y^2 - вибіркові дисперсії

Коефіцієнт вимірює лінійну кореляцію між двома змінними. Результуюче значення лежить в [-1; 1], де -1 означає ідеальну негативну кореляцію (коли одна змінна збільшується, інша зменшується), +1 означає ідеальну позитивну кореляцію і 0 означає відсутність лінійної кореляції між двома змінними. Зазвичай це швидкий і простий спосіб для підрахунку кореляції і часто являється однією з перших дій, яку варто виконати над даними.

Очевидним недоліком кореляції Пірсона як механізму класифікації характеристик є те, що він чутливий до лінійних взаємозв'язків. Якщо співвідношення нелінійне, кореляція Пірсона може бути близькою до нуля, навіть якщо є відповідність один до одного між двома змінними.

Ще одним методом являється лінійний дискримінантний аналіз (Linear discriminant analysis або LDA) який використовується для пошуку лінійної комбінації ознак, що характеризує або відокремлює два або більше класів (або рівнів). Дискримінантний аналіз є близьким до дисперсійного і регресійного

аналізів, які також намагаються виразити одну із залежних змінних у вигляді лінійної комбінації інших показників або вимірювань. Однак, у двох інших методів залежна змінна є числовий величиною, в той час як у дискримінантному аналізі це категорійна змінна.

Дисперсійний аналіз (Analysis of variance або ANOVA) - подібний до ЛДА, за винятком того, що він управляється за допомогою одного або декількох категоричних незалежних функцій та однієї неперервної функції. Він забезпечує статистичну перевірку того, чи є стани кількох груп рівними чи ні.

Також в цих цілях використовується критерій хі квадрат - статистичний тест що застосовується до груп категорійних ознак для оцінки вірогідності кореляції або асоціації між ними з використанням їх частотного розподілу. Вважається що критерій хі-квадрат - асимптотично вірний критерій, тобто розподіл вибірки можно зробити скільки завгодно близьким до розподілу хі-квадрат шляхом збільшення розміру вибірки.

Рекурсивне виключення характеристик

Даний метод (Recursive Feature Elimination або RFE) працює шляхом рекурсивного видалення атрибутів та побудови моделі на тих атрибутах, які залишаються. Він використовує значення точності моделі, щоб визначити, які атрибути (і комбінації атрибутів) найбільше сприяють прогнозуванню результуючого значення.

Спочатку, алгоритм навчається за початковим набором функцій, і важливість кожної характеристики визначається певним коефіцієнтом. Тоді найменш важливі характеристики будуть видалені з поточного набору. Ця процедура рекурсивно повторюється на зменшеному наборі, доки в кінцевому підсумку не буде досягнуто бажаної кількості обраних характеристик.

Перевага такого підходу полягає в тому, що він не видаляє змінні, які вважаються незначними на початку процесу, але стають все більш і більш значущими в подальшому, оскільки менш важливі ознаки видаляються. Для наборів даних з багатьма змінними, відносно сильно корельованих один з одним і порівняно слабко корелюючими з результуючою змінною, цей підхід може

призвести до дещо іншого вибору характеристик порівняно з тим, що робиться за наївним вибором на основі моделі. Недолік цього методу полягає в тому, що оскільки ви мусите багато разів тренувати модель, цей підхід значно повільніший від алгоритмів одноразового виконання[15].

Метод головних компонент

Метод головних компонент (Principal Component Analysis або PCA) використовує ортогональне перетворення множини спостережень з можливо пов'язаними змінними (сутностями, кожна з яких приймає різні числові значення) у множину змінних без лінійної кореляції. Як правило, це називається технікою скорочення даних. Його властивістю є те, що ви можете вибрати кількість розмірів або головних компонент у перетвореному результаті.

Серед інших подібних методів, що дозволяють узагальнювати значення елементарних ознак, МГК виділяється простою логічною конструкцією, й у той же час на його прикладі стають зрозумілими загальна ідея й цілі численних методів факторного аналізу.

Метод головних компонент дає можливість по m — числу вихідних ознак виділити r головних компонент, або узагальнених ознак. Простір головних компонент ортогональний. Математична модель методу головних компонент базується на логічному припущенні, що значення множини взаємозалежних ознак породжують деякий загальний результат.

Важливість характеристик

Для оцінки важливості характеристик можна використовувати дерева рішень, такі як "Випадковий ліс"(Random Forest) та "Додаткові дерева"(Extra Trees). Дані методи машинного навчання використовуються для класифікації, регресії та інших завдань, які оперують за допомогою побудови численних дерев прийняття рішень під час тренування моделі і продукують моду для класів (класифікацій) або усереднений прогноз (регресія) побудованих дерев.

Перший крок в оцінці важливості змінної в тренувальному наборі - тренування випадкового лісу на цьому наборі. Під час процесу побудови моделі для кожного елемента тренувального набору вважається так звана out-of-bag

помилка. Потім для кожної сутності така помилка опосередковується по всьому випадковому лісі.

Для того, щоб оцінити важливість j -ого параметра після тренування, значення j-ого параметра перемішуються для всіх записів тренувального набору та out-of-bag помилка рахується знову. Важливість параметра оцінюється шляхом усереднення по всіх деревах різниці показників out-of-bag помилок до і після перемішування значень. При цьому значення таких помилок нормалізуються на стандартне відхилення.

Параметри вибірки, які дають більші значення, вважаються більш важливими для тренувального набору. Таким чином, метод може надати числове значення важливості впливу кожної характеристики на класифікацію або прогнозування кінцевого результату.

Отже, хоча всі розглянуті методи для удосконалення набору характеристик для оптимізації методів машинного навчання і можуть в результаті давати схожі набори характеристик, однак вони відрізняються у певних тонкощах своєї роботи і надають різну додаткову інформацію, в залежності від наших потреб.

Висновки до розділу 2

В даному розділі було проведене дослідження основних методів багатофакторної автентифікації користувачів та особливостей їх використання. Ми дослідили методи двофакторної автентифікації та визначили що автентифікація основана на основі одноразового згенерованого паролю на основі часу буде найефективнішим рішенням при досягненні наших цілей.

Поглиблене дослідження поведінкових патернів користувачів у системах обміну миттєвими повідомленнями дозволило нам виділити важливі ключові ознаки, які можуть сприяти підвищенню точності роботи нашого механізму автентифікації вхідного повідомлення.

Також були розглянуті способи оптимізації та покращення точності класифікатора, досліджені методи удосконалення вибору підмножини вектору

ознак, такі як одномірний вибір, рекурсивне виключення характеристик, метод головних компонент та метод оцінки важливості характеристик. На основі порівняння було обрано метод рекурсивного виключення характеристик, що дозволяє підвищити точність нашого механізму, зменшити час тренування моделі та зниження її перенавчання. В наступному розділі, займемося реалізацією нашої системи з використанням двофакторної автентифікації та проведемо удосконалення існуючого рішення з урахуванням вищезгаданих методів.

3 РОЗРОБЛЕННЯ УДОСКОНАЛЕНОГО МЕТОДУ АВТЕНТИФІКАЦІЇ

Базуючись на дослідженнях проведених у попередніх розділах можемо прийти до висновку що для забезпечення підвищеного рівня захищеності і запобігання витоку інформації, тобто попередження про його ймовірність, наш механізм має володіти наступними властивостями:

- використовувати двофакторну автентифікацію користувачів на початку сесії обміну повідомленнями
- застосовувати удосконалений метод машинного навчання для автентифікації повідомлень
- забезпечувати зручний інтерфейс для обміну повідомленнями, їх аналізу та попередження про можливу загрозу

Задовольнити усім перерахованим вимогам ми можемо за допомогою бота побудованого на основі платформи Telegram, з реалізацією автентифікації за допомогою введення TOTP через один з додатків для їх генерації, можливістю запрошувати співрозмовника до приватного чату, автентифікації повідомлень та донавчання нашої моделі на основі побудованого методу машинного навчання.

3.1 Побудова захищеної системи обміну миттєвими повідомленнями

Розробка бота для нашого рішення буде відбуватися за допомоги мови Java, з використанням бібліотеки Telegram Bot Java Library[16] з імплементованим Telegram Bot API, для спрощення програмної розробки.

Основний функціонал необхідний для нашої задачі, це можливість авторизуватися в системі, підтвердити свою особистість за допомогою двофакторної автентифікації на основі TOTP, запросити співрозмовника до діалогу, з попередньою валідацією його особистості. Також, підключення нашого класифікатору для збору інформації про особливості відправки повідомлень,

тренування на цих даних та аналізу нових повідомлень на відповідність до поведінкового патерну користувача, оповіщення при виявленні ризику порушенні автентичності повідомлення.

3.2 Реалізація методу двофакторної автентифікації

Для реалізацію методу генерації одноразових паролів на основі часу також використовуватимемо мову Java. Запровадимо утилітний клас, в якому реалізуємо необхідні методи для автентифікації на основі TOTP:

- генерація секретного ключа
- генерація QR кода для сканування на основі цього ключа
- перевірка одноразового пароля згенерованого додатком користувача

Секретний ключ будемо генерувати в форматі base32 (літери A–Z, та цифри 2–7) використовуючи генератор рандомних чисел мови Java *java.security.SecureRandom*.

```
public static String generateSecret() {
    StringBuilder builder = new StringBuilder(SECRET_LENGTH);
    Random rnd = new SecureRandom();
    for (int i = 0; i < SECRET_LENGTH; i++) {
        int val = rnd.nextInt(32);
        builder.append( (val < 26) ? (char) ('A' + val) : (char) ('2' + (val - 26)));
    }
    return builder.toString();
}
```

Надалі реалізуємо метод для генерації QR кода, він використовуватиме спеціальний ключ, який ідентифікуватиме нашу програму, *String keyId* та згенерований раніше секретний ключ *String secret*:

```
public static String generateQRCodeURL(String keyId, String secret) {
    StringBuilder builder = new StringBuilder(128);
    builder.append("https://chart.googleapis.com/chart?chs=200x200&cht=qr&chl=200x200&chl=");
```

```
        builder.append("otpauth://totp/").append(keyId).append("%3Fsecret%3D").append(secret);
        return builder.toString();
    }
```

Після введення секретного ключа або сканування QR коду за допомогою спеціального додатку-автентифікатора на своєму мобільному пристрої, користувач матиме згенерований одноразовий пароль, який вводитиме в систему. Для його валідації реалізуємо наступний метод:

```
public static boolean validateTOTP(String secret, int userTOTP) {
        long timeMillis = System.currentTimeMillis();
        byte[] key = decodeBase32(secret);
        long totp = generateTOTP(key, timeMillis);
        return totp == userTOTP;
}
```

В середині він бере поточний системний час, розшифровує згенерований раніше ключ та генерує власний пароль на основі цих даних, який має бути рівний з паролем користувача. Для генерації коду автентифікації повідомлень використовуються методи бібліотека *javax.crypto* на основі HmacSHA1.

Тепер, для автентифікації користувача, нам залишається лише послідовно викликати вказані методи в нашій системі.

3.3 Удосконалення методу машинного навчання аналізу поведінкових патернів

Під час минулого дослідження нами було побудовано механізм виявлення повідомлень, що за характеристиками не відповідали поведінковим паттернам співрозмовника. Тренування та тестування моделі проводилося на діалогах різних користувачів. Для набору даних у якості легальних елементів були використані їх повідомлення, а за потенційно шкідливі елементи прийняті повідомлення інших

користувачів, тобто повідомлення з особливостями притаманними іншим користувачам.

3.3.1 Аналіз правопису користувачів при обміні поведінковими патернами

Застосування аналізу повідомлень на частоту порушення правил написання слів при обміні миттєвими повідомленнями. Для дослідження було обрано п'ять наборів даних з різним ступенем вираженості патерну користувачів. На початку до звичайного аналізу речення була додана перевірка на порушення одну з чотирьох загальних помилок при написанні слова: видалення букви, додавання зайвої, перестановка літер та підстановка (заміна правильної букви хибною).

Для перевірки правопису повідомлень та підрахунку помилок певного типу розробимо додатковий модуль за допомогою мови Python та розширемо існуючий алгоритм отриманими результатами. За основу для перевірки правильності написання слова візьмемо словник розміром 100000 слів. Враховуючи перераховані вище помилки, можемо здогадатися що виправлення слова, що не входить в словник, набуватимуть також чотирьох форм:

- додавання необхідної букви до слова
- видалення зайвої букви зі слова
- перестановка сусідніх літер місцями
- заміна літер

Також, варто врахувати що можливі випадки, коли у слові допущені дві помилки різних типів, будемо враховувати це як особливий тип помилок. На основі попереднього дослідження за використанням Баєсівського класифікатору ми отримали наступні результати:

Таблиця 3.1 – Результати роботи класифікатора для різних користувачів та різних розмірів вектору ознак

Vector size	user1	user2	user3	user4	user5
50	0.859	0.859	0.711	0.717	0.811
100	0.907	0.868	0.746	0.714	0.819
150	0.91	0.87	0.808	0.735	0.856
200	0.893	0.868	0.805	0.739	0.856
250	0.893	0.868	0.788	0.736	0.856
300	0.895	0.866	0.756	0.742	0.837
350	0.896	0.864	0.793	0.744	0.819
400	0.896	0.864	0.774	0.737	0.828
450	0.896	0.864	0.774	0.735	0.828

З урахуванням ознак перерахованих вище, модифікуємо нашу реалізацію та проведемо повторне тренування нашої моделі, маємо:

Таблиця 3.2 – Результати класифікатора з удосконаленим вектором ознак

Vector size	user1	user2	user3	user4	user5
50	0.92	0.889	0.811	0.806	0.819
100	0.962	0.889	0.819	0.789	0.828
150	0.949	0.903	0.856	0.783	0.875
200	0.935	0.902	0.856	0.779	0.875
250	0.947	0.885	0.856	0.783	0.876
300	0.922	0.885	0.837	0.788	0.857
350	0.91	0.885	0.819	0.781	0.84
400	0.91	0.885	0.828	0.767	0.849
450	0.91	0.885	0.828	0.767	0.849

За результатами дослідження середнє значення точності розпізнавання повідомлень. що не належать користувачеві, класифікатору з розширеним вектором ознак становить 85,7%, в порівнянні з 81,9% роботи попереднього алгоритму. Що підтверджує значущість особливостей правопису різних користувачів при аналізі їх поведінкових патернів. Таким чином, бачимо що враховуючи правила нам вдалося збільшити точність нашого алгоритму на 3,8%.

3.3.2 Вибір підмножини ознак характеристичного вектору

Для оптимізації нашого методу, а саме зменшення розміру вектору ознак, зниження часу тренування та підвищення точності ми можемо застосувати один з методів зменшення кількості характеристик (Feature Subset Selection або FSS). Нам важливо лише дізнатися чи входить певна ознака в множину найоптимальніших, а додаткові дані, такі як значення важливості цієї характеристики, нем не важливі. В такому випадку нам підійде метод рекурсивного виключення ознак, що за заданою кількістю необхідних характеристик, підбере для нас множину найважливіших ознак.

Оскільки на різних етапах спілкування ми матимемо різну величину характеристичного вектора, проведемо дослідження на декількох наборах даних та визначимо застосування якої підмножини характеристик буде найоптимальнішим. Результати дослідження для одного з набору даних приведені у наступній таблиці (3.3)

Таблиця 3.3 – Результати роботи класифікатора для різних розмірів вектору ознак з застосуванням удосконаленого вибору найважливіших ознак

Vector size	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
50	0,75	0,896	0,942	0,943	0,943	0,933	0,923	0,923	0,911	0,92
100	0,88	0,926	0,932	0,934	0,924	0,943	0,94	0,94	0,949	0,962
150	0,956	0,908	0,944	0,944	0,943	0,94	0,941	0,95	0,962	0,949
200	0,956	0,897	0,944	0,934	0,941	0,94	0,94	0,938	0,938	0,935
250	0,955	0,914	0,933	0,932	0,939	0,938	0,925	0,926	0,936	0,947
300	0,935	0,915	0,932	0,943	0,939	0,928	0,927	0,914	0,911	0,922
350	0,946	0,903	0,932	0,943	0,94	0,929	0,928	0,901	0,912	0,91

400	0,935	0,954	0,933	0,943	0,94	0,926	0,939	0,912	0,924	0,91
450	0,924	0,931	0,933	0,942	0,94	0,938	0,924	0,912	0,909	0,91
AVG	0,915	0,916	0,936	0,939	**0,939**	0,935	0,932	0,924	0,928	0,929

Таким чином, за середнім значенням точності класифікації бачимо, що показник у 0,5 від загальної кількості характеристик надає нам в середньому найточніші результати. Це означає, що приблизно половина наших характеристик, тобто слів, не відіграють значної ролі при класифікації повідомлення, що доволі справедливо враховуючи особливості нашої мови, відмінності словникових запасів між людьми та загальновживаними словами, які не несуть додаткової інформації.

3.3.3 Вибір оптимального розміру вектору ознак для аналізу вхідних повідомлень

Оскільки користувачеві необхідно надати єдине значення ймовірності того що отримане повідомлення належить його відправнику, нам необхідно прийти до висновку, який з розмірів характеристичного вектору може надати найточніше значення. Для цього проведемо фінальне дослідження наших результатів, та визначимо оптимальну кількість характеристик для автентифікації повідомлення.

Таблиця 3.4 – Результати роботи удосконаленого класифікатора

Vector size	user1	user2	user3	user4	user5	AVG
50	0,943	0,847	0,925	0,813	0,819	0,8778
100	0,924	0,855	0,926	0,815	0,828	0,8764
150	0,943	0,843	0,925	0,827	0,875	0,8946
200	0,941	0,843	0,942	0,836	0,875	**0,8992**
250	0,939	0,841	0,942	0,832	0,876	0,8948
300	0,939	0,824	0,925	0,83	0,857	0,8872
350	0,94	0,824	0,907	0,823	0,84	0,879

400	0,94	0,838	0,907	0,823	0,849	0,8808
450	0,94	0,824	0,891	0,817	0,849	0,8764

Тобто, в середньому, після зростання характеристичного вектору більше за 200 ознак, точність класифікації починає зменшуватися, при тому що складність обчислень та ресурси, такі як час та пам'ять, які на них витрачаються збільшуються. Таким чином, для роботи нашого механізму можемо обрати дане значення за основне при тренуванні та аналізі вхідних повідомлень.

3.4 Аналіз отриманих результатів

Після застосування перевірки правопису вхідних повідомлень та врахування цих значень у векторі ознак для їх аналізу ми встановили що середня точність роботи нашої класифікації збільшилася на 3,8%. Також, бачимо що найкращим розміром вектору ознак, для якого точність роботи нашого класифікатору на різних наборах даних найбільша дорівнює 200. Оскільки, застосування методу рекурсивного видалення ознак показало при коефіцієнті 0,5 для вибору підмножини найважливіших ознак ми можемо отримати значення близьке, і навіть краще, в порівнянні з використанням повного вектору, та враховуючи інші переваги які він нам надає (зменшення часу тренування та розпізнавання, перешкоджання перенавчанню), оберемо його для фінального тестування.

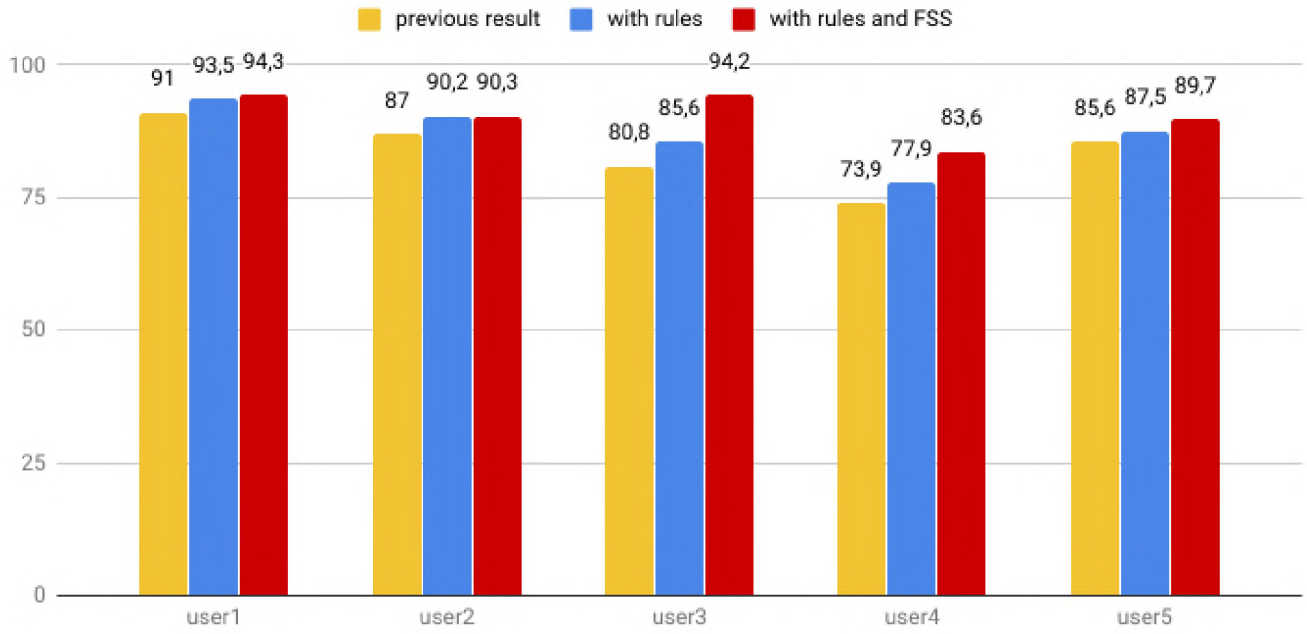

Рисунок 3.1 – Результати порівняння точності роботи класифікатору на різних наборах даних в ході удосконалення

Таким чином, нам вдалося суттєво збільшити точність роботи нашого алгоритму, а саме на 6,7% в середньому, оптимізувати його роботу завдяки вибору найбільш суттєвих ознак.

та розробити платформу з використанням двофакторної автентифікації на основі TOTP для зручного та безпечного обміну повідомленнями, що значно знизить загрози взлому облікового запису та витоку інформації.

3.5 Практичне застосування

Тепер, коли нам вдалося суттєво покращити роботу нашого алгоритму автентифікації вхідних повідомлень користувачів, підключимо його до нашої системи обміну миттєвими повідомленнями та надамо необхідний інтерфейс для спілкування. На початку розмови користувач має автентифікуватися в системі. Для цього, після натиснення кнопки “Start” нашого бота, для користувача згенерується особистий секретний ключ та QR код на його основі. Після

додавання облікового запису з цим ключем і QR кодом до додатку генерації одноразових паролей у користувача з'явиться 6 цифровий пароль, що буде оновлюватися кожні 30 секунд. Приклад QR коду згенерованого нашою системою можемо побачити на рисунку 3.2:

Рисунок 3.2 – QR код згенерований за таємним ключем

Після введення цього коду користувач матиме можливість запросити співрозмовника через команду *"/invite username"*, де *username* - унікальний логін користувача в Telegram, на основі чого згенерується запрошення у вигляді посилання у вигляді "https://telegram.me/SecureAuthChatBot?start=*invitation_hashcode*", яке він має надіслати цій людині. Після чого вона має подібним чином пройти автентифікацію та продовжити розмову. Надалі інформація про повідомлення користувачів буде передаватися на наш механізм та враховуватиметься при тренуванні моделі. Кожне повідомлення класифікуватиметься та, при наявності певного ризику, сповіщатиме користувача про можливе порушення автентичності вхідного повідомлення, що дозволятиме користувачеві бути обачнішим при наданні важливої інформації та запобігатиме потенційному витоку інформації.

Висновки для розділу 3

Таким чином, в результаті роботи було розроблено систему для обміну миттєвими повідомленнями з впровадженням додаткових рівнів безпеки у вигляді автентифікації на основі TOTP та застосування нашого механізму класифікації вхідних повідомлень на основі Баєсового класифікатору з вдосконаленим вибором вектору ознак.

Тестування удосконаленого механізму продемонструвало що нам вдалося суттєво збільшити точність роботи нашого алгоритму, а саме на 6,7% в середньому, що являється чудовим результатом на діалогах різних користувачів, з унікальними особливостями ведення переписки, та з урахуванням незначної кількості повідомлень для деяких них. Отриманої точності у 89% цілком достатньо для попередження користувача про можливість порушення автентичності повідомлення і досягнення основної мети роботи, тобто запобігання витоку інформації при обміні повідомленнями.

Розроблена система на основі чатботу платформи Telegram надає можливість використовувати реалізовані методи автентифікації на практиці, з використанням зручного інтерфейсу, та забезпечує додатковий рівень безпеки, що стає суттєвим методом перевірки особистості співрозробника та автентифікації його повідомлень, що в значній мірі дозволяє знизити ризик витоку конфіденційної інформації.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	*Напрямки застосування*	*Вигоди для користувача*
Впровадження двофакторної автентифікації за допомогою TOTP та автентифікації повідомлень співрозмовника за їх поведінковими патернами при обміні повідомленнями.	1. Підключення до популярних систем обміну миттєвими повідомленнями	Впровадження додаткового рівня захищеності при обміні повідомленнями
	2. Створення власного продукту на основі розробленого механізму	Забезпечення захисту від загроз витоку інформації завдяки удосконаленим методам автентифікації

Розроблене рішення не можна порівнювати з іншими, оскільке воно унікальне в своєму роді та не має конкурентів. Основою для нього являється принцип дії класифікаторів спаму, однак для систем обміну миттєвими повідомленнями. Рішення застосовується у ролі додатку до існуючих систем, відповідає за автентифікацію користувачів та повідомлень, а не за передачу даних, тому, можемо лише оцінити його техніко-економічні характеристики, сильні сторони та недоліки.

Таблиця 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко - економічні характеристики ідеї	Мій проект	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
1.	Економічні	Витрати на розробку рішення, закупку ліцензій, розміщення на хостингу, маркетинг - 20000$	Витрати на сервери зі збільшенням користувачів	Немає	Масштабованість, експоненційний рост користувачів, реферальна система
2.	Технічні	Використання посиленої автентифікації користувачів та повідомлень	Збільшення кількості етапів перед початком спілкування	Немає	Адаптованість до будь-якої платформи, широке коло користувачів
3.	Надійності	Впровадження додаткового рівня автентифікації користувачів	Передачею повідомлень займаються платформи передачі повідомлень	Немає	Удосконалена автентифікація користувачів, впевненість користувачів у автентичності повідомлень
4.	Технологічні	Підключення до існуючих систем обміну повідомленнями	Немає	Немає	Удосконалення механізму автентифікації повідомлень зі збільшенням набору даних

Продовження таблиці 4.2

5.	Ергономічні	Первинна автентифікація користувачів	Немає	Використання додатків автентифікаторів	Мінімум додаткових зусиль для початку роботи
6.	Естетичні	Інтерфейс системи обміну повідомленнями	Немає	Зовнішній вигляд обумовлений інтерфейсом основної платформи	Зручний інтерфейс використання рішення
7.	Екологічність	Обслуговування серверної частини додатку	Немає	Використання платних серверів надійних служб	Немає

4.2 Технологічний аудит ідеї проекту

Оскільки наше рішення являється програмним продуктом, основними технологіями будемо розглядати існуючі бібліотеки та SDK для спрощення розробки.

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1.	Надлаштування для систем обміну повідомленнями	SDK конкретної платформи на мові Java, наприклад для Telegram	Офіційна документація платформи	Доступна
2.	Автентифікація вхідних повідомлень	Реалізація TOTP	Використання існуючих алгоритмів	Доступна
3.	Автентифікація користувачів	Реалізація баєсового класифікатору мовою Python	Викоирстання бібліотек scilearn мови Python	
Обрана технологія реалізації ідеї проекту: всі технології для реалізації ідеї проекту наявні та доступні				

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Оскільки при плануванні впровадження проекту на ринок ми будемо інтегруватися з існуючими популярними системами обміну повідомленнями, конкурентних рішень ми не матимемо.

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	6

Продовження таблиці 4.4

2	Загальний обсяг продаж, грн/ум.од	200 млн ум. од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Підтримка інтеграцій з ботами платформою
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	Близько 200%

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1	Безпечна передача інформації в системах обміну миттєвими повідомленнями	Будь який користувач месенджерів, тобто майже всі користувачі мережі інтернет	Основна цільова група користувачів продукту - ділові партнери, бізнесмени, люди що володіють потенційно важливою інформацією. Друга група - прості користувачі, які бажають підвищити рівень безпеки при передачі інформації	- висока точність автентифікації повідомлень - утримання інформації в конфіденційному у стані

Таблиця 4.6 - Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1	Недовіра користувачів системам обміну миттєвими повідомленнями	Зменшення кількості користувачів визване порушеннями конфіденційності інформації або її використання тощо	Відсутність впливу на існуючі рішення призведе до потреби у створення власного мессенджеру
2	Обмеження роботи додатків у месенджерах	Заборона або суттєве обмеження повноважень додатків, неможливість продовження відповідного функціонування	Прийняття нових правил роботи або створення власного рішення

Таблиця 4.7 - Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*
1	Почастішання випадків витоку інформації	Зростаюча потреба користувачів в надійному способі попередити подібні загрози	Розширення клієнтської бази, маркетингові дії
2	Покращення рівня підтримки додаткових інтеграцій месенджерами	Сприяння компаній систем обміну миттєвими повідомленнями розробці та інтеграції нових рішень	Збільшення масштабів розвитку

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції: Монополія	Заборона або суттєве обмеження повноважень додатків, неможливість продовження відповідного функціонування	Необхідність прийняття нових правил роботи або створення власного рішення
2. За рівнем конкурентної боротьби: Глобальний	Витіснення або обмеження компаніями додатків з своїх систем	Створення власного рішення
3. За галузевою ознакою Внутрішньогалузева	Рішення застосовується в одній галузі	Розширення сфери надання послуг та функціоналу
4. Конкуренція за видами товарів: - товарно-видова	Рішення використовується для задоволення потреб клієнтів, але істотно відрізняються від рішень конкурентів на користь якості існуючих сервісів	Впровадження рішення в дрібні системи обміну повідомленнями, корпоративні чати
5. За характером конкурентних переваг - нецінова	Вартість не є ключовим фактором для клієнтів	Покращення якості продукту та збільшення функціоналу збільшує клієнтську базу
6. За інтенсивністю - марочна	Результуюча привабливість продукту	Підвищення якості роботи механізму

Таблиця 4.9 - Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари замінники*
Складові аналізу	*Навести перелік прямих конкурентів: Відсутні*	*Визначити бар'єри входження в ринок: Signal, Telegram*	*Визначити фактори сили постачальників: Не впливають*	*Визначити фактори сили споживачів: Відмова від користування продуктом або ж навпаки визнання*	*Фактори загроз з боку замінників: Повернення до листування, email, дзвінків*
Висновки	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів: Відсутня	Можливе блокування просування нашого рішення	Чи постачальники диктують умови роботи на ринку? Які? Не диктують умови	Чи клієнти диктують умови роботи на ринку? Які?: Вимоги до якості продукту, точності його роботи	Обмеження для роботи на ринку через товари замінники: Не передбачається

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
1	Відсутність прямих конкурентів	Унікальність розробленого рішення, використання додаткових рівнів автентифікації
2	Багатоплатформеність	Впровадження автентифікації в існуючі популярні системи обміну миттєвими повідомленнями
3	Доступність	Простота в використанні клієнтами, велика кількість користувачів

Продовження таблиці 4.10

4	Зручність	Зручний інтерфейс у використанні, обумовлений інтерфейсом основної платформи
5	Багатофункціональність	Застосування підходить для різної цільової аудиторії

Порівняльний аналіз сильних та слабких сторін “MultiAuth”.

Оскільки прямих конкурентів немає, ми не можемо визначити відносний рейтинг нашого рішення при використанні в існуючих системах обміну повідомленнями. Зазначимо лише, що воно надасть додатковий рівень захищеності при спілкуванні шляхом автентифікації користувачів та їх повідомлень.

Таблиця 4.11 - SWOT-аналіз стартап-проекту

Сильні сторони: - практична корисність продукту - відсутність конкурентів - незалежність від політично економічного стану країни	Слабкі сторони: - залежність від політики компаній розробників IMS - потреба у використанні користувачами додатків автентифікаторів для TOTP
Можливості: - підтримка впровадження рішення більшою кількістю IMS - сприяння популярними компаніями IMS застосуванню нашого рішення, інтеграція його в основний функціонал - збільшення інтересу користувачів до підвищення безпеки обміну повідомленнями через певні фактори	Загрози: - обмеження функціональності платформами IMS - втрата довіри до IMS або рішення за певних обставин

Таблиця 4.12 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1	Повноцінна реалізація для однієї платформи	Дуже висока	3-6 місяця
2	Підключення до основних платформ	Висока	1 рік
3	Розробка власної платформи на основі існуючого рішення та залучення клієнтів	Ймовірна, при успішному впровадженні попередніх рішень	2 роки

Отже, з перелічених альтернатив, найкращим рішенням буде початок роботи з повноцінної реалізації продукту для однієї платформи, та поступовим розширенням і переходом до багатьох платформ виділенням власної.

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.13 - Вибір цільових груп потенційних споживачів

№ п/п	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1	Звичайні користувачі	Користувачі зацікавлені в безпеці обміну повідомленнями готові його використовувати	Середній, попит у користувачів, що обмінюються важливою інформацією	Відсутня	Простий вхід, необхідність проведення маркетингових компаній

Продовження таблиці 4.13

2	Компанії та підприємства з корпоративними IMS	Компанії, що турбуються про обмін інформації серед працівників будуть його використовувати	Високий рівень попиту	Відсутня	Можливі складнощі з зміною політики компаній щодо впровадження продукту
3	Компанії IMS	Впровадження додаткового рівня безпеки може бути цікавим для компаній	Низький рівень попиту, через малу кількість ключових гравців	Відсутня	Складність в переконанні та підтвердженні корисності рішення
Які цільові групи обрано: звичайні користувачі та компанії та підприємства з корпоративними IMS					

За результатами аналізу потенційних груп споживачів будемо використовувати стратегію диференційованого маркетингу.

Таблиця 4.14 - Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку*
1	Повноцінна реалізація для однієї платформи	Стратегія диференційованого маркетингу	Підвищення рівня захищеності обміну повідомленнями	Стратегія диференціації

Продовження таблиці 4.14

2	Підключення до основних платформ	Стратегія диференційованого маркетингу	Розширення ринку продукту	Стратегія диференціації
3	Розробка власної платформи на основі існуючого рішення та залучення клієнтів	Стратегія диференційованого маркетингу	Виділення власного продукту з усіма перевагами у використанні	Стратегія спеціалізації

Таблиця 4.15 - Визначення базової стратегії конкурентної поведінки

№ п/п	*Чи є проект «першопроходцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки*
1	Так	Рішення буде доступне всім користувачам існуючих платформ	Ні	Стратегія лідера, розширення первинного попиту

Таблиця 4.16 - Визначення стратегії позиціонування

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
1	Забезпечення додаткового рівня безпеки при обміні повідомленнями; перевірка автентичності співрозмовника на початку бесіди та автентифікація повідомлення	Стратегія диференціації	Надання додаткового рівня безпеки при обміні повідомленнями в обраній IMS	Багатофакторна автентифікація користувача Сповіщення про можливе порушення автентифікації повідомлення Безпечне використання існуючих IMS

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.17 - Визначення ключових переваг концепції потенційного товару

№ п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
1	Безпечний обмін повідомленнями	Впровадження додаткового рівня безпеки при обміні повідомленнями шляхом перевірки автентичності співрозмовника на початку бесіди та автентифікації повідомлення	Додаткова автентифікація співрозмовника та надісланих повідомлень

Таблиця 4.18 - Опис трьох рівнів моделі товару

Рівні товару	*Сутність та складові*		
I. Товар за задумом	Забезпечення додаткового рівня безпеки при обміні повідомленнями, перевірка автентичності співрозмовника на початку бесіди та автентифікація повідомлення		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Використання багатофакторної автентифікації користувачів	М	Тх, Е
	2. Автентифікація повідомлень	М	Тх, Е
	3. Інтеграція з популярними IMS	М	Тх, Тл, Е
	4. Можливість налаштовувати рівень автентичності повідомлень	М	Тл, Е
	Якість: RFC 6238(TOTP), RFC 4226 (HOTP)		
	Марка: MultiAuth		
III. Товар із підкріпленням	До продажу: використання безкоштовної версії з обмеженим функціоналом		
	Після продажу: надання послуг без обмежень, проведення додаткового маркетингу		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності, унікальний підхід в автентифікації повідомлень			

Таблиця 4.19 - Визначення меж встановлення ціни

№ п/п	*Рівень цін на товари-замінники*	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
1	Відсутній	Відсутній	Будь-який	Для клієнтів 5у.о. на місяць; для компаній 40/300/500у.о на місяць

Таблиця 4.20 - Формування системи збуту

№ п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
1	Потреба в встановленні додаткового рівня безпеки при обміні повідомленнями	Підтримка користувачів, оновлення програмного рішення	Прямий канал між компанією та користувачами	Прямі продажі

Таблиця 4.21 - Концепція маркетингових комунікацій

№ п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій цільових клієнтів*	*Ключові позиції, обрані для позиціонування*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*

1	Тестування продукту, якості автентифікації повідомлень, вплив на рівень захищеності ІМ	Системи обміну миттєвими повідомленнями	Введення додаткового рівня захищеності ІМ, запобігання витоку інформації	Вказати на можливі загрози при ІМ та запропонувати вирішення цих проблем	Демонстрація актуальних загроз при ІМ та їх вирішення за допомогою нашого продукту

Висновки до розділу 4

В результаті проведеного аналізу бачимо, що проект має непогані можливості ринкової комерціалізації. Має перспективи впровадження з огляду на потенційні групи клієнтів, а саме користувачів систем обміну поівдомленнями та компанії з корпоративними месенджерами. Бар'єри входження полягають лише в готовності користувачами використовувати дане рішення та залежать від відношення компаній розробників IMS у його сприянню. Конкуренція для нашого рішення відсутня.

Для впровадження нашого рішення, найкраще почати з інтеграції з однією з найпопулярніших систем обміну повідомленнями, та поступово проводити інтеграції з іншими системами. В перспективі, при зростанні клієнтської бази можливе виділення в окрему систему, тому подальша імплементація проекту є доцільною.

ВИСНОВКИ

Під час виконання дипломної роботи ми розглянули рівень захищеності сучасних систем обміну миттєвими повідомленнями, дослідили методи багатофакторної автентифікації користувачів та автентифікації повідомлень на основі виділення ключових характеристик притаманних їх відправникам.

Ми побудували систему для безпечного обміну повідомленнями за допомогою автентифікації на основі одноразових згенерованих паролей на початку діалогу та подальшій автентифікації повідомлень з використанням баєсівського класифікатору. Було удосконалено набір ключових ознак для тренування обраної моделі, оптимізовано вибір найсуттєвіших характеристик для аналізу вхідних повідомлень, що дозволило зменшити час тренування, відкинути шуми з даних та збільшити точність класифікації.

За результатами тестування, попри впровадження додаткового рівня безпеки всієї системи, було збільшено показник точності класифікації в середньому на 6,7% при точності роботи алгоритма до 94,3%, у порівнянні з кращими результатом минулого дослідження 89,2%.

Розроблена система може використовуватися як незалежний продукт для безпечного обміну повідомленнями між людьми, з запобіганням доступу неавторизованих користувачів до обміну повідомленнями, тобто заволодіння конфіденційною інформацією, на етапах початку бесіди, а також впродовж обміну повідомленнями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1 Most popular mobile messaging apps worldwide as of October 2018, based on number of monthly active users (in millions) [Електронний ресурс] – Режим доступу до ресурсу: https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/

2 Global Information Assurance Certification Paper [Електронний ресурс] / C. Sweigart – Режим доступу до ресурсу: https://www.giac.org/paper/gsec/2917/instant-messaging-security/103935

3 A Secure Instant Messaging System [Електронний ресурс] / M. Wrzesinska – Режим доступу до ресурсу: http://students.mimuw.edu.pl/SR/prace-mgr/wrzesinska/thesis6.html

4 Modern Phishing Campaigns And Effective Prevention [Електронний ресурс] / C. Goldschmidt [Електронний ресурс] – 2018.– Режим доступу до ресурсу: https://www.forbes.com/sites/forbestechcouncil/2018/03/02/modern-phishing-campaigns-and-effective-prevention/#23d424fd649d

5 The Privacy Problems with Mobile Messaging Apps [Електронний ресурс] / L. Caetano [Електронний ресурс] – 2014. – Режим доступу до ресурсу: https://securingtomorrow.mcafee.com/consumer/mobile-and-iot-security/viber-app-sends-data-unencrypted/

6 A Survey on Cyber Security Evolution and Threats: BiometricAuthentication Solutions. In Biometric Security and Privacy; Springer: Berlin, Germany [Текст] / Benarous, L.; Kadri, B.; Bouridane, A. – 2017; 371–411 с.

7 Multi-Factor Authentication: A Survey [Електронний ресурс] / A. Ometov, S. Bezzateev, N. Mäkitalo – 2018 – Режим доступу до ресурсу: https://www.researchgate.net/publication/322288752_Multi-Factor_Authentication_A_Survey

8 Why You Shouldn't Use SMS for Two-Factor Authentication (and What to Use Instead) [Електронний ресурс] / C. Hoffmann – 2017 – Режим доступу до ресурсу: https://www.howtogeek.com/310418/why-you-shouldnt-use-sms-for-two-factor-authentication/

9 TOTP: Time-Based One-Time Password Algorithm [Електронний ресурс] / D. M'Raihi, S. Machani. M. Pei, – 2011. – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc6238

10 HOTP: An HMAC-Based One-Time Password Algorithm [Електронний ресурс] / D. M'Raihi, S. Machani. M. Pei, – 2005. – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc4226

11 The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed [Текст] / Hastie, T., Tibshirani R., Friedman J.., 2009. – 764 с.

12 An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages Proceeding of the an International ACM SIGIR Conference on Res and Devel in Inform Retrieval [Текст] / I. Anderouysopoulos, J. Koutsias, K.V. Chandrianos, G. Paliouras, C. Spyrpolous. – 2000. – 465 с.

13 Practical Machine Learning Tools and Techniques (second ed.) [Текст] / Ian H. Witten, Eibe Frank., 2005. – 558 с

14 How to Write a Spelling Corrector [Електронний ресурс] / P. Norvig, – 2017. – Режим доступу до ресурсу: http://norvig.com/spell-correct.html

15 A comparative study of feature selection methods for stress hotspot classification in materials [Електронний ресурс] / A. Mangal, E. Holm - 2018 Режим доступу до ресурсу: https://arxiv.org/pdf/1804.09604.pdf

16 Telegram Bot Java Library [Електронний ресурс] / R. Bermudez, – 2016. – https://monsterdeveloper.gitbooks.io/writing-telegram-bots-on-java/content/chapter1.html

ДОДАТКИ

Додаток А

Реалізація двофакторної автентифікації користувачів за допомогою TOTP через бота для платформи Telegram.

Основний клас проекту

```
@SpringBootApplication
public class BotApplication {

    {
        ApiContextInitializer.init();
    }

    public static void main(String[] args) {
        SpringApplication.run(BotApplication.class);
    }

    @Autowired
    private TelegramBot telegramBot;

    @Bean
    public TelegramBotsApi botsApi() {
        TelegramBotsApi botsApi = new TelegramBotsApi();

        try {
            botsApi.registerBot(telegramBot);
        } catch (TelegramApiException e) {
            e.printStackTrace();
        }
        return botsApi;
    }
}
```

Сервіс для телеграм боту

```
@Service
public class TelegramBot extends TelegramLongPollingBot {

    @Autowired
    private UserService userService;

    Map<String, String> inviteUsers = new HashMap<>();

    @Override
    public void onUpdateReceived(Update update) {
        BotUser user = userService.createIfNotExist(update.getMessage().getFrom());

        String messageText = update.getMessage().getText();

        process(user, messageText);
    }

    @Override
    public String getBotUsername() {
        return "SecureAuthChatBot";
```

```
}

@Override
public String getBotToken() {
    return "794203158:AAGH5-WyKNrc_Q7KyhOLXLXj8fODhiBszT4";
}


public void process(BotUser user, String messageText) {
    if (messageText.startsWith("/start") && messageText.split(" ").length > 1) {
        if (!processInvitedMember(user, messageText)) {
            return;
        }
    }

    if (messageText.startsWith("/invite")) {
        inviteMember(user, messageText);
        return;
    }

    switch (user.getStatus()) {
        case UserStatus.NEW:
            sendRegistrationInfo(user);
            break;
        case UserStatus.REGISTERED:
        case UserStatus.EXPIRED:
            processTOTPOrSendMessage(user, messageText);
            break;
        case UserStatus.AUTHENTICATED:
            sendMessageToPartner(user, messageText);
            break;
        default:
            sendDefaultMessage(user);
    }
}

private boolean processInvitedMember(BotUser user, String messageText) {
    boolean isValid = false;
    String invitationKey = messageText.split(" ",2)[1];
    System.out.println("User " + user.getUserName() + " invited with key " + invitationKey);
    if (inviteUsers.get(invitationKey).equals(user.getUserName())) {
        Integer chatId = Integer.valueOf(invitationKey.split("AAA", 2)[1]);
        BotUser partner = userService.findByChatId(chatId);
        sendMessage(user, "Welcome to secure chat with " + partner.getUserName() + " !");
        userService.updatePartnerId(user, chatId);
        userService.updatePartnerId(partner, user.getChatId());
        inviteUsers.remove(invitationKey);
        isValid = true;
    } else {
        sendMessage(user, "Something went wrong.");
    }
    return isValid;
}

private void inviteMember(BotUser user, String messageText) {
    String invitationKey = UUID.randomUUID().toString() + "AAA" + user.getChatId();
    String invitedUser = messageText.split(" ", 2)[1];
    System.out.println("User " + user.getUserName() + " invites " + invitedUser);
    inviteUsers.put(invitationKey, invitedUser);
    sendMessage(user,"https://telegram.me/SecureAuthChatBot?start=" + invitationKey);
}

private void sendRegistrationInfo(BotUser user) {
```

```
    sendMessage(user, "Set up mfa with this key: " + user.getSecret() + "\n" +
            "Or scan QR code: " + TOTPUtil.generateQRCodeURL("diploma", user.getSecret()));
    userService.changeStatus(user, UserStatus.REGISTERED);
}

private void processTOTPOrSendMessage(BotUser user, String messageText) {
    System.out.println("Process TOTP " + messageText + " for user " + user.getUserName());
    boolean isValidated = false;
    if (messageText.length() == 6 && isNumeric(messageText)) {
        try {
            isValidated = TOTPUtil.validateTOTP(user.getSecret(), Integer.valueOf(messageText));
        } catch (GeneralSecurityException e) {
            e.printStackTrace();
        }
        if (isValidated) {
            userService.changeStatus(user, UserStatus.AUTHENTICATED);
            if (user.getPartnerId() != null) {
                BotUser partner = userService.findByChatId(user.getPartnerId());
                messageText = "Ok, send your messages to " + partner.getUserName();
            } else {
                messageText = "Ok, invite your friend to chat! Enter `/invite username` to do it.";
            }
        } else {
            messageText = "Not valid: " + messageText;
        }
    } else {
        messageText = "Not secure: " + messageText;
    }
    sendMessage(user, messageText);
}

private boolean isNumeric(String text) {
    boolean isNumeric = false;
    try {
        Integer.parseInt(text);
        isNumeric = true;
    } catch (Exception e) {
        System.out.println("Not valid number: " + text);
    }
    return isNumeric;
}

private void sendMessage(BotUser user, String messageText) {
    send(textMessage(user.getChatId(), messageText));
}

private void sendMessageToPartner(BotUser user, String messageText) {
    send(textMessage(user.getPartnerId(), messageText));
}

private SendMessage textMessage(Integer chatId, String messageText) {
    return new SendMessage()
            .setChatId(chatId.toString())
            .setText(messageText);
}

private void sendDefaultMessage(BotUser user) {
    send(textMessage(user.getChatId(), "What?"));
}

private void send(SendMessage message) {
    System.out.println("Send message: " + message.getText() + " to " + message.getChatId());
    try {
```

```
            execute(message);
        } catch (TelegramApiException e) {
            e.printStackTrace();
        }
    }
}
```

Сервіс для збереження даних користувача в базу даних

```
@Service
public class UserService {

    @Autowired
    private UserRepository userRepository;

    public BotUser findByChatId(Integer chatId) {
        return userRepository.findByChatId(chatId);
    }

    public BotUser createIfNotExist(User user) {
        BotUser botUser = userRepository.findByChatId(user.getId());
        if (botUser == null) {
            botUser = new BotUser();
            botUser.setChatId(user.getId());
            botUser.setFirstName(user.getFirstName());
            botUser.setLastName(user.getLastName());
            botUser.setUserName(user.getUserName());
            botUser.setStatus(UserStatus.NEW);
            botUser.setSecret(TOTPUtil.generateSecret());
            return userRepository.save(botUser);
        }
        return botUser;
    }

    public BotUser changeStatus(BotUser user, String status) {
        user.setStatus(status);
        return userRepository.save(user);
    }

    public BotUser updatePartnerId(BotUser user, Integer chatId) {
        user.setPartnerId(chatId);
        return userRepository.save(user);
    }
}
```

Додаток Б

Удосконалення механізму автентифікації повідомлень за поведінковими патернами користувача.

```
def make_Dictionary(train_file, ratio):
    file = open(train_file)
    end = int((sum(1 for line in file)) * ratio)

    print train_file, ' size ', end

    all_words = []
    for i, line in enumerate(open(train_file)):
        if (i == end):
            break
        line = line[line.index('M:') + 2:].lower()

        words = re.findall(r"[\w']+|[0-9.,!?;)(:]+", line)

        all_words += words

    dictionary = Counter(all_words)

    print "dictionary done, len ", len(dictionary)


    return dictionary

def extract_features(train_file, start, end):
    dialogues_number = sum(1 for line in open(train_file))

    if (start == 0):
        end = int(dialogues_number * end)
        values_number = end
    else:
        start = int(dialogues_number * start)
        values_number = dialogues_number - start
    features_matrix = np.zeros((values_number, features_number))


    file = open(train_file)
    for i,line in enumerate(file):
        if (i < start):
            continue
        if (i == end):
            break
        i = i - start
        data = line.split(' ', 2)


        # set time category
        if with_timer:
            time_category = getTimeCategoty(int(data[1][data[1].index(':') + 1:]))
        else:
            time_category = getTimeCategoty(data[1][data[1].index(':') + 1:])
        features_matrix[i, time_category] = 1
```

```
        # extract message words features
        message = data[2][data[2].index(':') + 1:].lower()

        words = re.findall(r"[\w']+|[0-9.,!?;:)(]+", message)

        #set broken rule
        if with_rules:
            broken_rules = getBrokenRule(words)
            for ib, broken_rule in enumerate(broken_rules):
                if broken_rule != 0:
                    features_matrix[i, int(ib + 8)] = 1



        for word in words:
            for j,d in enumerate(dictionary):
                if d[0] == word:
                    wordID = j
                    features_matrix[i,wordID + add_features] = 1   # words.count(word)
    return features_matrix

def getBrokenRule(words):
    result = findWordsRule(words)
    for i, res in enumerate(result):
        if res != 0:
            result[i] = 1
    return result

def getTimeCategoty(responseTime):
    if (responseTime == 1):
        return 1
    elif (responseTime > 1 and responseTime <= 15):
        return 2
    elif (responseTime > 15 and responseTime <= 45):
        return 3
    elif (responseTime > 45 and responseTime <= 90):
        return 4
    elif (responseTime > 90 and responseTime <= 180):
        return 5
    elif (responseTime > 180 and responseTime <= 300):
        return 6
    elif (responseTime > 300 and responseTime <= 900):
        return 7
    else:
        return 0

def extract_message_features(newMessage):
    features_matrix = np.zeros((1, features_number))

    data = newMessage.split(' ', 1)

    #time category
    time_category = getTimeCategoty(data[0][data[0].index(':') + 1])
    features_matrix[0, int(time_category)] = 1

    message = data[1][data[1].index(':')+1:].lower()

    words = re.findall(r"[\w']+|[0-9.,!?;)(]+", message)

    for word in words:
        for j,d in enumerate(dictionary):
            if d[0] == word:
```

```
                wordID = j
                features_matrix[0,wordID + time_groups_number] = 1   # words.count(word)
    return features_matrix

def extract_labels(train_file, start, end):
    dialogues_number = sum(1 for line in open(train_file))

    if (start == 0):
        end = int(dialogues_number * end)
        target_values = [None] * end
    else:
        start = int(dialogues_number * start)
        target_values = [None] * (dialogues_number - start)

    for i,line in enumerate(open(train_file)):
        if (i < start):
            continue
        if (i == end):
            break
        i = i - start

        data = line.split(' ', 2)

        target_values[i] = data[0][data[0].index(':')+1:]

    return target_values

def getAccuracy(test_labels, result):
correct = 0
for x in range(len(test_labels)):
            if test_labels[x] == result[x]:
                        correct += 1
return (correct/float(len(test_labels)))


def getFileResults(dataset, dictionary_size):
    global features_number, dictionary
    features_number = dictionary_size + add_features

    dictionary = full_dictionary.most_common(dictionary_size)

    train_matrix = extract_features(dataset, 0, ratio)
    train_labels = extract_labels(dataset, 0, ratio)

    clf = BernoulliNB()
    clf.fit(train_matrix, train_labels)

    test_matrix = extract_features(dataset, ratio, 0)
    test_labels = extract_labels(dataset, ratio, 0)

    result = clf.predict(test_matrix)

    # print 'dictionary used: ', dictionary_size

    confusion_matrix_values = confusion_matrix(test_labels, result)
    fn = confusion_matrix_values[0][1]
    tn = confusion_matrix_values[0][0]
    fp = confusion_matrix_values[1][0]
    tp = confusion_matrix_values[1][1]


    precision = round(float(tp) / (tp + fp), 2)
    recall = round(float(tp) / (tp + fn), 3)
```

```
    accuracy = round(getAccuracy(test_labels, result), 2)

    # print dictionary_results[user]
    dictionary_results[user][dictionary_size] = recall

    accuracy = round(getAccuracy(test_labels, result), 2)

    if (accuracy > accuracy_results[dataset][1][1]):
        accuracy_results[dataset][1] = [dictionary_size, accuracy]
    if (accuracy < accuracy_results[dataset][0][1]):
        accuracy_results[dataset][0] = [dictionary_size, accuracy]



def main():
    global ratio, time_groups_number, full_dictionary, user, rules_number, add_features
    ratio = 8. / 10
    time_groups_number = 8
    rules_number = 6

    add_features = time_groups_number + (rules_number if with_rules else 0)

    user = 0

    train_dir = 'dataset'
    datasets = [os.path.join(train_dir, f) for f in os.listdir(train_dir)]
    for dataset in datasets:
        dictionary_results[user] = {}
        full_dictionary = make_Dictionary(dataset, ratio)
        accuracy_results[dataset] = [[0, 100], [0, 0]]
        for i in xrange(50, 500, 50):
            if i > len(full_dictionary):
                break
            getFileResults(dataset, i)

        print collections.OrderedDict(sorted(dictionary_results[user].items()))
        print dataset
        for key in sorted(dictionary_results[user]):
            print dictionary_results[user][key]
        user += 1
    user = 0

with_rules = True

dictionary_results = [None] * 5
accuracy_results = {}
main()
```